



TESIS - KI142502

**PENINGKATAN AKURASI ESTIMASI USAHA DAN
BIAYA PERANGKAT LUNAK PADA *COCOMO II*
BERDASARKAN MODEL *LOGIKA FUZZY GAUSSIAN*
DAN *BEE COLONY OPTIMIZATION***

RAHMI RIZKIANA PUTRI
5115201012

DOSEN PEMBIMBING
Daniel Oranova Siahaan, S.Kom., M.Sc., PDEng
NIP. 197411232006041001

Sarwosri, S.Kom., M.T.
NIP. 19768092001122001

PROGRAM MAGISTER
BIDANG KEAHLIAN REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

[Halaman ini sengaja dikosongkan]



THESIS - KI142502

INCREASING THE ACCURACY OF ESTIMATE EFFORT AND SOFTWARE COST ON COCOMO II MODEL BASED FUZZY GAUSSIAN LOGIC AND BEE COLONY OPTIMIZATION

RAHMI RIZKIANA PUTRI
5115201012

SUPERVISOR
Daniel Oranova Siahaan, S.Kom., M.Sc., PDEng
NIP. 19741123 200604 1 001

Sarwosri, S.Kom., M.T.
NIP. 19760809 200112 2 001

MASTER PROGRAM
DEPARTEMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

[Halaman ini sengaja dikosongkan]

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:

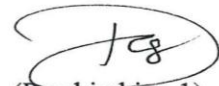
RAHMI RIZKIANA PUTRI
Nrp. 5115201012

Dengan judul:
PENINGKATAN AKURASI ESTIMASI USAHA DAN BIAYA PERANGKAT LUNAK
PADA *COCOMO II* BERDASARKAN MODEL LOGIKA *FUZZY GAUSSIAN* DAN *BEE COLONY OPTIMIZATION*

Tanggal Ujian: 19 Juni 2017
Periode Wisuda: 2016 Genap

Disetujui oleh:

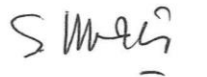
Daniel Oranova Siahaan, S.Kom, M.Sc, PD.Eng.
NIP. 197411232006041001


(Pembimbing 1)

Sarwosri, S.Kom., M.T.
NIP. 197608092001122001


(Pembimbing 2)


Dr. Ir. Siti Rochimah, M.T
NIP. 196810021994032001


(Penguji 1)

Rizky Januar Akbar, S.Kom, M.Eng
NIP. 198701032014041001


(Penguji 2)

Fajar Baskoro, S.Kom, M.T
NIP. 197404031999031002


(Penguji 3)


Dekan Fakultas Teknologi Informasi,
Dr. Agus Zamal Arifin, S. Kom., M.Kom.
NIP. 197208091995121001

[Halaman ini sengaja dikosongkan]

PENINGKATAN AKURASI ESTIMASI USAHA DAN BIAYA PERANGKAT LUNAK PADA *COCOMO II* BERDASARKAN MODEL *LOGIKA FUZZY GAUSSIAN* DAN *BEE COLONY OPTIMIZATION*

Nama Mahasiswa : Rahmi Rizkiana Putri
NRP : 5115201012
Pembimbing : Daniel Oranova Siahaan,S.Kom.,M.Sc.,PDEng
Sarwosri,S.Kom.,M.T.

ABSTRAK

Pengembangan perangkat lunak merupakan proses yang tidak sepenuhnya sempurna. Masih sering terdapat kegagalan hingga ditolaknya suatu proyek perangkat lunak. Salah satu faktor penting yang mempengaruhi keberhasilan proyek yaitu perkiraan usaha dan biaya. Perkiraan usaha dan biaya yang akurat akan memberikan manajemen yang baik untuk proyek perangkat lunak. Apabila perkiraan usaha dan biaya kurang akurat maka akan mempengaruhi manajemen proyek perangkat lunak dan kurang efektifnya proses pengembangan proyek tersebut. Dalam beberapa dekade terakhir industry pembuatan perangkat lunak telah diperkenalkan dengan model estimasi COCOMO II. Penambahan *cost driver* COCOMO II yang diperkenalkan Barry Boehm pada tahun 2000 digunakan dalam penulisan ini guna memberikan hasil akurasi yang lebih baik karena telah mencakup keseluruhan bagian yang di estimasi. Namun berdasarkan penelitian, akurasi hasil perkiraan usaha dan biaya dengan metode *COCOMO II Fuzzy Gaussian* masih jauh dari *Actual Effort*.

Oleh sebab itu peningkatan akurasi dari hasil *COCOMO II Fuzzy Gaussian* masih dapat dilakukan dengan menggunakan metode *Bee Colony Optimization* yang dapat menghasilkan optimasi yang lebih baik, terlihat dari hasil MMRE loyal. Selain itu penggunaan metode *Bee Colony Optimization* dapat meningkatkan akurasi yang dihasilkan, meminimalkan error antara estimasi biaya dengan nilai yang sebenarnya. Penulisan ini tidak hanya sebatas menerapkan metode *Bee Colony Optimization* saja, tetapi juga melakukan perubahan nilai parameter A dan B pada *COCOMO II* dengan gradual awal adalah 0,01 untuk mencapai nilai optimal pada gradual tertentu. Perubahan nilai pada parameter A dan B dilakukan dengan cara menaikkan dan menurunkan rentang nilai dengan gradual yang telah ditentukan. Apabila hasil optimal telah didapatkan maka pencarian total nilai kesalahan (MMRE loyal) dihentikan.

Berdasarkan hasil implementasi dari metode yang diusulkan pada penulisan ini, kesalahan akurasi perkiraan usaha dan biaya proyek perangkat lunak dapat turun 38% bila dibandingkan penelitian sebelumnya. Dengan demikian metode yang diusulkan membuktikan bahwa kesalahan dalam melakukan perkiraan usaha dan biaya perangkat lunak semakin berkurang dan mendekati dengan nilai yang sebenarnya. Sehingga, akurasi estimasi usaha dan biaya proyek perangkat lunak dapat ditingkatkan.

Kata Kunci: Akurasi, Bee Colony Optimization (BCO), Constructive Cost Model II (COCOMO II), Fuzzy Gussian, Perkiraan Biaya Proyek Perangkat Lunak, Usaha.

[Halaman ini sengaja dikosongkan]

INCREASING THE ACCURACY OF ESTIMATE EFFORT AND SOFTWARE COST ON COCOMO II MODEL BASED FUZZY GAUSSIAN LOGIC AND BEE COLONY OPTIMIZATION

Student Name : Rahmi Rizkiana Putri
NRP : 5115201012
Supervisor : Daniel Oranova Siahaan,S.Kom.,M.Sc.,PDEng.
Sarwosri,S.Kom.,M.T.

ABSTRACT

Software development is a process that is not completely perfect. There are still frequent failures until the rejection of a software project. One of the important factors affecting project success is business and cost estimation. Accurate business and cost estimates will provide good management for software projects. If the business and cost estimates are less accurate then it will affect the management of the software project and the ineffectiveness of the project development process. In the last few decades software industry has been introduced with COCOMO II estimation model. The addition of the COCOMO II cost driver introduced by Barry Boehm in 2000 is used in this paper to provide better accuracy results as it covers the entire section in the estimation. However, based on the research, the accuracy of business and cost estimates using COCOMO II Fuzzy Gaussian method is still far from Actual Effort.

Therefore, the increased accuracy of COCOMO II Fuzzy Gaussian results can still be done using Bee Colony Optimization method which can result in better optimization, seen from MMRE loyal results. In addition, the use of the Bee Colony Optimization method can improve the resulting accuracy, minimizing the error between estimated cost and actual value. This is not only to apply the Bee Colony Optimization method, but also to change the values of parameters A and B on COCOMO II with the initial gradual is 0.01 to achieve optimal value on a certain gradual. The change of values on parameters A and B is done by raising and lowering the range of values with a predetermined gradual. If optimal results have been obtained then the total search error value (MMRE loyal) is stopped.

Based on the results of the implementation of the proposed method at this writing, the accuracy error of business estimates and software project costs may fall by 38% when compared to previous studies. Thus the proposed method proves that mistakes in making business estimates and software costs are diminishing and closer to their true value. Thus, the accuracy of business estimates and software project costs can be increased.

Keywords: Accuracy, Bee Colony Optimization, Constructive Cost II model (COCOMO II), Effort, Estimates Software Project Cost, Fuzzy Gaussian.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tesis yang berjudul “Peningkatan Akurasi Estimasi Usaha Dan Biaya Perangkat Lunak Pada COCOMO II Berdasarkan Model Logika Fuzzy Gaussian Dan Bee Colony Optimization” sesuai dengan target waktu yang diharapkan.

Pengerjaan Tesis ini merupakan suatu kesempatan yang sangat berharga bagi penulis untuk belajar memperdalam ilmu pengetahuan. terselesaikannya buku Tesis ini tidak terlepas dari bantuan dan dukungan semua pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Tesis ini dengan baik.
2. Kedua orang tua penulis yang selalu mendoakan agar selalu diberikan kelancaran dan kemudahan dalam menyelesaikan Tesis ini. Serta menjadi motivasi terbesar untuk mendapatkan hasil yang terbaik.
3. Bapak Daniel Oranova Siahaan, S.Kom., M.Sc., PDEng dan Ibu Sarwosri, S.Kom., M.T. selaku dosen pembimbing yang telah memberikan kepercayaan, motivasi, bimbingan, nasehat, perhatian serta semua bantuan yang telah diberikan kepada penulis dalam menyelesaikan Tesis ini.
4. Ibu Dr. Ir. Siti Rochimah, M.T., Bapak Rizky Januar Akbar, S.Kom, M.Eng, dan Bapak Fajar Baskoro, S.Kom., M.Kom selaku dosen penguji yang telah memberikan bimbingan, saran, arahan, dan koreksi dalam pengerjaan Tesis ini.
5. Bapak Waskitho Wibisono, S.Kom., M.Eng., PhD selaku ketua program pascasarjana Teknik Informatika ITS, Bapak Dr.Eng. Darlis Herumurti, S.Kom, M.Sc selaku dosen wali penulis dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Mbak Lina, Mas Kunto dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
7. Seluruh keluarga besar yang selalu memberi semangat, doa, dukungan kepada penulis.
8. Rekan-rekan angkatan 2015 Pasca Sarjana Teknik Informatika ITS yang telah menemani dan memberikan bantuan serta motivasi untuk segera menyelesaikan Tesis ini.

9. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu disini yang telah membantu terselesaikannya Tesis ini.

Sebagai manusia biasa, penulis menyadari bahwa Tesis ini masih jauh dari kesempurnaan dan memiliki banyak kekurangan. Sehingga dengan segala kerendahan hati, penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, Juli 2017

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Penelitian	3
1.3. Perumusan Masalah	4
1.4. Batasan Masalah	4
1.5. Manfaat Penelitian	4
1.6. Kontribusi Penelitian	4
BAB 2 KAJIAN PUSTAKA	5
2.1. Penelitian Terkait	5
2.2. Software Project	5
2.3. Software Cost Estimation	6
2.4. COCOMO II	7
2.5. Fuzzy Gaussian	9
2.5.1. Triangular Membership Function	10
2.5.2. Trapezoidal Membership Function	10
2.5.3. Gaussian Membership Function	11
2.6. Bee Colony Optimization	12
BAB 3 METODOLOGI PENELITIAN	15
3.1. Tahapan Penelitian	15
3.2. Studi Literatur	15
3.3. Analisis dan Perancangan	16
3.3.1 Analisis	16
3.3.2 Perancangan Cost Driver COCOMO II	24
3.3.3 Memodelkan Fuzzy Logic Ke Dalam COCOMO II	28
3.3.4 Perancangan COCOMO II Fuzzy Gaussian Dengan BCO	31
3.4 Implementasi Metode	31

3.5.	Uji Coba dan Evaluasi.....	33
3.5.1	Uji Coba	33
3.5.2	Evaluasi	35
BAB 4	UJI COBA DAN EVALUASI	37
4.1.	Implementasi Penelitian	37
4.2.	Perancangan Uji Coba.....	37
4.2.1.	Pembagian Dataset	37
4.3.	Skenario Uji Coba	38
4.4.	Implementasi COCOMO II Fuzzy Gaussian	39
4.5.	Implementasi Bee Colony Optimization Pada COCOMO II.....	40
4.6.	Hasil Uji Coba Skenario 1.....	43
4.7.	Hasil Uji Coba Skenario 2.....	44
4.8.	Analisis Hasil	44
BAB 5	PENUTUP	49
5.1.	Kesimpulan	49
5.2.	Saran.....	50
DAFTAR PUSTAKA		51
LAMPIRAN		55
BIODATA PENULIS		63

DAFTAR GAMBAR

Gambar 2. 1. Model COCOMO II	8
Gambar 2. 2. Triangular Membership Function	10
Gambar 2. 3. Trapezoidal Membership Function	10
Gambar 2. 4. Gaussian Membership Function.....	11
Gambar 2. 5. Fuzzy Logic System dengan Fuzzifier dan Defuzzifier	12
Gambar 3. 1. Tahap Penelitian.....	15
Gambar 3. 2. Rancangan Cost Driver COCOMO II.....	25
Gambar 3. 13. Alur Proses Implementasi BCO	32
Gambar 3. 14. Nilai Effort Multiplier APEX	34
Gambar 4. 1. Grafik MMRE dataset Cocomo-sdr	47
Gambar 4. 2. Grafik MMRE loyal dataset Nasa 93-dem.....	47

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Model pengembangan dengan karakteristik proyek	7
Tabel 3.1 Nilai Effort Multiplier COCOMO II	17
Tabel 3.2 Nilai Scale Factor COCOMO II	17
Tabel 3.3 Deskripsi RELY	18
Tabel 3.4 Deskripsi DATA	18
Tabel 3.5 Deskripsi RESU	19
Tabel 3.6 Deskripsi DOCU	19
Tabel 3.7 Deskripsi TIME	19
Tabel 3.8 Deskripsi STOR	20
Tabel 3.9 Deskripsi PVOL	20
Tabel 3.10 Deskripsi ACAP	20
Tabel 3.11 Deskripsi PCAP	21
Tabel 3.12 Deskripsi PCON	21
Tabel 3.13 Deskripsi APEX	21
Tabel 3.14 Deskripsi PLEX	22
Tabel 3.15 Deskripsi LTEX	22
Tabel 3.16 Deskripsi TOOL	22
Tabel 3.17 Deskripsi SITE	23
Tabel 3.18 Deskripsi SCED	23
Tabel 3.19 Deskripsi CPLX	23
Tabel 3.20 Effort Multiplier Fuzzy Gaussian	34
Tabel 4.1 TIME Descriptor	40
Tabel 4.2 Cocomo-sdr Dengan Gradual 0,01	41
Tabel 4.3 Cocomo-sdr Dengan Gradual 0,001	42
Tabel 4.4 Cocomo-sdr Dengan Gradual 0,0001	42
Tabel 4.5 Nasa 93-dem Dengan Gradual 0,01	42
Tabel 4.6 Nasa 93-dem Dengan Gradual 0,001	43
Tabel 4.7 Nasa 93-dem Dengan Gradual 0,0001	43
Tabel 4.8 MMRE Dengan Metode Cocomo II Fuzzy Gaussian	44
Tabel 4.9 MMRE loyal Dengan Metode Bee Colony Optimization	44

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Pengembangan perangkat lunak merupakan proses yang tidak sepenuhnya sempurna. Di dalamnya sering terdapat keterlambatan, kegagalan, hingga ditolaknya suatu proyek perangkat lunak. Sebagai contoh perkiraan, manajemen, efisiensi biaya dan usaha seluruh proyek, hal tersebut adalah faktor – faktor penting dalam pengukuran perkiraan proyek perangkat lunak. Manajemen yang baik dari pengembangan perangkat lunak dapat memperkirakan biaya dan sumber daya perangkat lunak yang tepat. Perkiraan biaya perangkat lunak menyediakan jalur yang penting antara konsep umum dan teknik analisis ekonomi dan khususnya rekayasa perangkat lunak. Perkiraan biaya proyek adalah tugas memprediksi biaya, usaha atau produktivitas yang dibutuhkan untuk menyelesaikan proyek (Eberendu, 2014). Dalam perkiraan proyek perangkat lunak, pentingnya untuk menyeimbangkan hubungan antara usaha, jadwal, kualitas, yang mana merupakan tiga aspek utama (Borade & Khalkar, 2013).

Di perangkat lunak, terdapat reputasi yang tidak baik mengenai perkiraan biaya (Sharma, 2011). Proyek besar perangkat lunak cenderung memiliki frekuensi yang sangat tinggi, baik dari kelebihan jadwal yang sudah ditentukan, kelebihan biaya, masalah kualitas, dan pembatalan langsung. Meskipun mendapat reputasi buruk, tetapi beberapa proyek besar perangkat lunak dapat selesai tepat waktu, tetap sesuai anggaran, dan beroperasi dengan sukses ketika digunakan (Sharma, 2011). Di tesis ini bertujuan untuk fokus terhadap masalah perkiraan biaya dalam proyek perangkat lunak. Ada terdapat beberapa teknik atau metode yang terkenal dalam hal perkiraan biaya perangkat lunak, salah satunya adalah COCOMO (C. S. Reddy & Raju, 2009a). Pada tahun 2000 Barry Boehm memperkenalkan COCOMO II yang merupakan pengembangan dari COCOMO dan telah terbukti lebih akurat dengan beberapa perbaikan di beberapa *cost driver*, dan akan digunakan dalam penelitian ini. Metode ini bertujuan untuk membantu bagaimana caranya membuat perkiraan biaya sebuah proyek perangkat lunak yang sederhana, mengurai langkah

– langkah dasar, istilah, dan perlengkapan yang digunakan. COCOMO II memperjelas tidak hanya dibagian biaya proyek yang diharapkan dan durasinya, tetapi juga memverifikasi semua sisi dasar proyek perangkat lunak dengan menyediakan metodologi, yang diuji pada berbagai proyek kehidupan yang nyata dan dengan demikian mengurangi resiko dan dapat membantu memberikan alasan yang masuk akal ketika berkomunikasi dengan *stakeholder* proyek (Sharma, 2011).

Cost driver proyek perangkat lunak yang dikembangkan secara karakteristik jelas dan pasti pada tahap awal dari siklus hidupnya. Karenanya sulit untuk menghasilkan upaya perkiraan yang akurat. Bias biasanya muncul ketika pengukuran *cost driver* perangkat lunak didasarkan pada penilaian manusia. Pendekatan ini tidak mempertimbangkan fitur yang jelas dan pasti yang ditempati dalam *cost driver*. Ketidakjelasan *cost driver* secara signifikan mempengaruhi akurasi perkiraan usaha dan biaya yang berasal dari model estimasi perangkat lunak. Karena ketidakjelasan dan ketidakpastian *cost driver* perangkat lunak tidak dapat dihindari (C. S. Reddy & Raju, 2009b) serta terkadang kurangnya informasi tentang model proyek, metode COCOMO II ini gagal untuk bekerja di pekerjaan yang tepat. Untuk mengatasi hal tersebut maka penelitian ini mengusulkan perlu dilakukan pendekatan *fuzzy* (Stimulation, 2013). Model *fuzzy* memiliki keuntungan untuk kemudahan memverifikasi *cost driver* dengan mengadopsi kumpulan *fuzzy*. Beberapa peneliti telah melaporkan kemajuan yang dibuat mengenai keberhasilan penerapan teknik logika *fuzzy* dalam membangun model estimasi usaha dan biaya perangkat lunak untuk meningkatkan metode (C. S. Reddy & Raju, 2009a). Untuk memperkirakan biaya berdasarkan logika *fuzzy*, akan lebih tepat digunakan ketika informasi yang di dapat tidak jelas dan tidak tepat (C. H. S. Reddy & Raju, 2009). Banyak fungsi keanggotaan yang terdapat dalam logika *fuzzy*. Agar mendapatkan akurasi yang lebih tepat dan mengurangi kesalahan dari COCOMO II, maka tesis ini menggunakan *fuzzy Gaussian*.

Fungsi keanggotaan *fuzzy Triangular* dan *fuzzy Trapezoidal* yang digunakan dalam COCOMO untuk menggantikan kuantisasi konvensional dengan menggunakan nilai interval *fuzzy*. Setelah mempelajari perilaku, untuk mendapatkan cara transisi interval yang baik maka penelitian ini menggunakan *fuzzy Gaussian*. Kinerja *fuzzy Gaussian* lebih baik dari anggota *fuzzy* lainnya,

karena menunjukkan transisi interval yang halus serta hasil yang dicapai lebih dekat dengan upaya atau usaha yang sebenarnya (C. S. Reddy & Raju, 2009a).

Sejumlah besar model perkiraan biaya telah tersedia namun hanya sedikit yang bisa mencapai tingkat kepuasan. COCOMO terkenal dapat memberikan hasil yang positif, namun karena meningkatnya kompleksitas dalam kebutuhan perangkat lunak, maka hasilnya menjadi kurang tepat. Perlu adanya metode optimasi, yaitu *Bee Colony Optimization* untuk mengoptimalkan COCOMO dari sisi model parameter konstan dan evaluasi MMRE untuk memeriksa kinerja yang lebih baik dan harus lebih minimum (Chalotra, Sehra, Brar, & Kaur, 2015).

Adapun metode yang digunakan pada penulisan ini adalah *fuzzy* dan optimasi. *Fuzzy* dengan *Gaussian Membership Function* akan digunakan untuk mencari nilai baru pada *effort multiplier* untuk meningkatkan kesesuaian nilai untuk setiap data *effort multiplier*, sedangkan *Bee Colony Optimization* untuk mencari parameter baru yang cocok pada perhitungan *estimate effort* untuk mendekati nilai *actual effort* serta mengoptimalkan hasil yang dicapai oleh COCOMO II.

Pada penelitian ini disusun sebagai berikut: Bagian 2 menjelaskan model COCOMO II dan ide umum dari metode yang digunakan dalam makalah ini. Bagian 3 secara ringkas menjelaskan kerja terkait dilakukan untuk memperkirakan upaya melalui logika *fuzzy* yang berbeda. Bagian 4 menyajikan model estimasi usaha *fuzzy* menggunakan fungsi *Gaussian* dan *Bee Colony Optimization* yang digunakan dalam penelitian ini. Bagian 5 menyajikan metodologi eksperimental dan hasil. Bagian terakhir menyimpulkan bahwa keakuratan estimasi usaha dapat ditingkatkan melalui model yang diusulkan dan upaya diperkirakan bisa sangat dekat dengan upaya yang sebenarnya.

1.2. Tujuan Penelitian

Penelitian ini bertujuan untuk menghasilkan peningkatan akurasi pada perkiraan biaya perangkat lunak dan memberikan kontribusi untuk mengurangi *error* antara biaya atau usaha yang sebenarnya dengan biaya atau usaha yang diperkirakan. Tujuan dari metode kontribusi ini adalah dapat meningkatkan akurasi COCOMO II yang menggunakan *fuzzy Gaussian* dengan mengoptimalkan hasil

MRE dan MMRE menggunakan *Bee Colony Optimization*. Hasil MMRE loyal diharapkan dapat meningkatkan minimal 16% dari penelitian sebelumnya.

1.3. Perumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengembangkan model perkiraan usaha dan biaya pada COCOMO II dengan menambahkan fungsi *Fuzzy Gaussian* dan *Bee Colony Optimization*?
2. Bagaimana meminimalkan *error* agar hasil usaha dan biaya antara estimasi dengan biaya sebenarnya memiliki sedikit selisih sehingga dapat meningkatkan akurasi?

1.4. Batasan Masalah

1. Dataset PROMISE Turkish Software Industry terdiri dari 12 proyek perangkat lunak.
2. Dataset PROMISE Nasa 93-dem terdiri dari 93 proyek perangkat lunak.

1.5. Manfaat Penelitian

Metode yang diusulkan diharapkan dapat mengetahui perkiraan biaya untuk proyek perangkat lunak, meminimalkan atau mengurangi *error* yang terdapat antara biaya atau usaha sebenarnya dengan biaya atau usaha yang diperkirakan. Serta mendapatkan hasil yang lebih akurat melalui metode yang diusulkan.

1.6. Kontribusi Penelitian

Kontribusi pada tesis ini adalah mengusulkan metode *Bee Colony Optimization* dengan mengubah nilai parameter A dan parameter B untuk mengoptimalkan hasil akurasi estimasi dari metode COCOMO II *Fuzzy Gaussian*. Hal tersebut berguna untuk mengurangi kesalahan antara hasil akurasi COCOMO II *Fuzzy Gaussian* dengan usaha dan biaya *actual effort*.

BAB 2

KAJIAN PUSTAKA

2.1. Penelitian Terkait

Terdapat penelitian yang terkait dengan topik ini, diantaranya yaitu yang pertama dilakukan oleh (C. H. S. Reddy & Raju, 2009). Pada penelitian tersebut, implementasi terhadap metode yang memiliki kinerja lebih baik, karena transisi yang baik dalam interval. Penelitian tersebut menggunakan metode fungsi keanggotaan *Fuzzy Gaussian* dengan dataset COCOMO. Menurut penelitiannya jika menggunakan *Fuzzy trapesium*, transisi yang dihasilkan tidak sebaik *Fuzzy Gaussian* sehingga hasil yang dicapai masih jauh dari *Actual Effort*. Metode yang diusulkan oleh peneliti (C. H. S. Reddy & Raju, 2009) menghasilkan MMRE 17,02%. Penelitian kedua dilakukan oleh (Savalgi, Jain, Andanappa, & Narayan, 2016) untuk menganalisa dampak dari upaya menghitung menggunakan berbagai fungsi keanggotaan dan ukuran kesalahan untuk estimasi usaha perangkat lunak. Penelitian tersebut menggunakan 8 macam keanggotaan *fuzzy* dan COCOMO II. Estimasi usaha menggunakan PSIGMF (*Gaussian Membership Function*) memberikan hasil yang lebih baik dibandingkan dengan fungsi keanggotaan lainnya. Adapun peneliti ketiga yaitu (My, 2016) mengusulkan metode COCOMO II-TLABC dengan dataset Nasa 93, menghasilkan MMRE 33,22%.

2.2. Software Project

Proyek perangkat lunak merupakan suatu proyek yang sangat berbeda jika dibandingkan dengan proyek-proyek lainnya, karena memiliki tujuan untuk menghasilkan produk yang jelas, berwujud, dan dapat dipahami. Tingkat kompleksitas manajemen proyek perangkat lunak lebih dari proyek-proyek lain. Fakta ini membuat siklus produksi menjadi begitu rumit dan sulit dalam proyek perangkat lunak. Ada begitu banyak alasan ketika proyek perangkat lunak tidak dapat diselesaikan tepat waktu (Khatibi, 2012) yaitu saat manajer proyek perangkat lunak dihadapkan dengan produksi yang tidak pasti dan tidak stabil yang sulit untuk dikontrol. Selain itu, kebutuhan pelanggan dan pengembangan teknologi serta alat-alat yang berubah dengan cepat di bidang ini. Terdapat pula masalah

komunikasi dapat menjadi faktor dalam keterlambatan dan kegagalan proyek perangkat lunak (Potdar & Potdar, 2014). Semua ini membuat prediksi upaya pembangunan menjadi sulit dalam proyek perangkat lunak. Sebagai solusi, menganalisis faktor yang efektif pada estimasi upaya pengembangan dapat meringankan permasalahan yang ada. Investigasi atribut proyek, keterbatasan, isu-isu manajemen dan pengetahuan pengembang dapat berguna untuk menarik kesimpulan dalam hal faktor yang efektif pada manajemen estimasi usaha dalam proyek perangkat lunak. Upaya prediksi adalah proses yang menantang dan rumit dalam proyek perangkat lunak. Ada beberapa atribut standar yang ditetapkan untuk setiap proyek perangkat lunak, yang meliputi jenis organisasi, jenis pembangunan, teknik pengembangan, gaya pengembangan, jenis aplikasi, bahasa pemrograman, alat yang digunakan serta ukuran (Khatibi, 2012). Ini adalah fakta bahwa keberhasilan proyek perangkat lunak tergantung pada perkiraan biaya yang dilakukan pada tahap awal proses pengembangan perangkat lunak. Sebuah survei terbaru mengatakan bahwa lebih dari 60% dari proyek perangkat lunak gagal karena berbagai alasan. Maka dari itu, dengan diterapkannya prinsip – prinsip manajemen proyek perangkat lunak yang tepat diharapkan proyek dapat diselesaikan tepat waktu dengan kualitas yang baik (L.Dasari, M.A.Kavya, V.Jasti, 2015).

2.3. Software Cost Estimation

Perkiraan biaya perangkat lunak adalah proses memprediksi jumlah usaha yang dibutuhkan untuk membangun sistem perangkat lunak. Perkiraan biaya perangkat lunak merupakan kegiatan terus menerus yang dapat dimulai pada tahap pertama dari siklus hidup perangkat lunak dan melalui waktu yang ditentukan (Borade & Khalkar, 2013). Sebagai akibat tidak adanya konsistensi dan kepastian, perkiraan tidak harus sistematis dan tepat sehingga beberapa perbaikan yang dibuat dalam upaya memperkirakan biaya harus menjadi prestasi besar dan itu berharga untuk perusahaan perangkat lunak (Stimation, 2013). Dalam dekade terakhir, banyak teknik estimasi usaha software telah diusulkan untuk mengevaluasi kinerja estimasi mereka (C. S. Reddy & Raju, 2009a). Beberapa teknik ini banyak digunakan termasuk estimasi oleh ahli, estimasi berbasis analogi, metode algoritmik, aturan induksi, jaringan syaraf tiruan dan logika *fuzzy* (Stimation, 2013).

Software metode estimasi biaya dibagi dalam dua kategori: metode algoritma dan metode non algoritmik (Waghmode & Kolhe, 2014).

2.4. COCOMO II

Umumnya COCOMO berlaku untuk 3 kelas proyek perangkat lunak. Model pengembangan karakteristik proyek dapat di lihat seperti di bawah ini dan tabel 2.1.

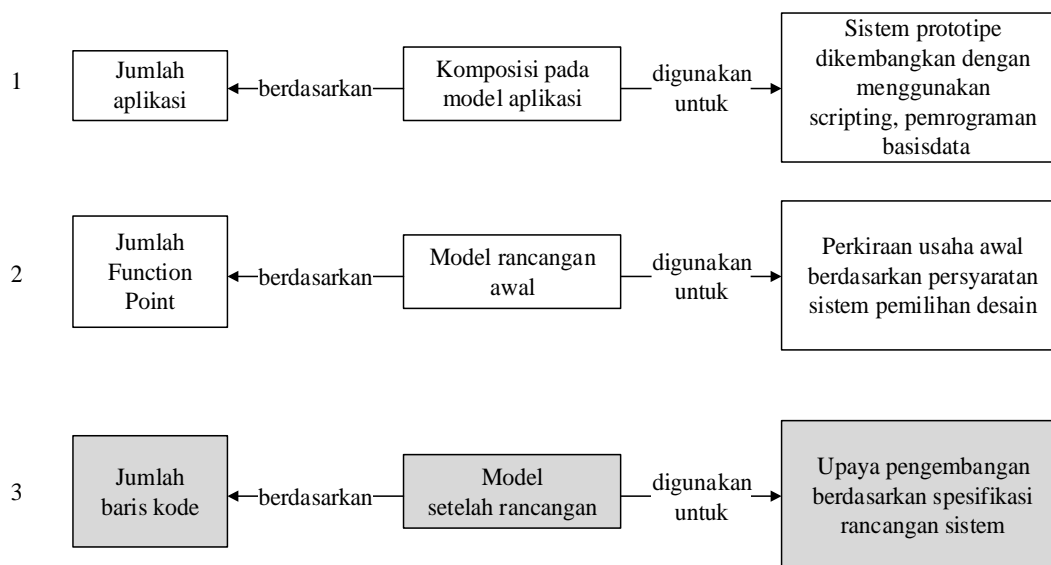
- a) Proyek organik – tim “kecil” dengan pengalaman bekerja yang “baik” dengan kebutuhan yang kurang “ketat” (Malik, Pandey, & Kaushik, 2013) selain itu proyek organik termasuk proyek yang relatif kecil, mirip dengan proyek-proyek yang dikembangkan sebelumnya, membutuhkan sedikit inovasi (Jaglan, 2016).
- b) Proyek semi – detached – tim “menengah” dengan beragam pengalaman kerja dengan kebutuhan yang kurang ketat (Malik et al., 2013) proyek ini termasuk juga dalam skala intermediate jika dalam ukuran dan kompleksitas (Jaglan, 2016).
- c) Proyek terintegrasi – Dikembangkan dalam sekumpulan hambatan yang ketat. Kombinasi antara proyek organik dan semi-detached (perangkat keras, perangkat lunak, operasional) (Malik et al., 2013).

Tabel 2.1 Model pengembangan dengan karakteristik proyek

	Size	Inovasi	Batas waktu	Lingkungan pengembangan
Organik	Kecil	Sedikit	Tidak ketat	Stabil
Semi detached	Menengah	Sedang	Sedang	Sedang
Terintegrasi	Besar	Lebih besar	Ketat	Perangkat keas yang rumit

Dengan mengikuti sistem klasifikasi Boehm ini, metode ini diringkas dalam tiga kategori (C. S. Reddy & Raju, 2009a) yaitu penilaian ahli, estimasi algoritmik, dan estimasi berdasarkan analogi. Metode algoritma didasarkan pada model matematika yang menghasilkan perkiraan biaya sebagai fungsi dari sejumlah variabel, yang dianggap sebagai faktor biaya utama (Borade & Khalkar, 2013). Menggunakan perkiraan biaya perangkat lunak metrik untuk mengidentifikasi teknik estimasi biaya, seperti sebagai SLOC (sumber baris kode). SLOC adalah metrik estimasi biaya lama yang mendapatkan popularitas di masa awal teknologi komputer, memberikan keuntungan lebih dari metrik estimasi biaya lainnya. Beberapa biaya, jadwal, dan usaha model estimasi menggunakan SLOC sebagai

parameter masukan, termasuk konstruktif biaya model banyak digunakan COCOMO seri model Boehm, 1981 (Eberendu, 2014) model algoritma seperti COCOMO II (Borade & Khalkar, 2013). COCOMO II yang diperkenalkan pada tahun 1994 untuk mengatasi masalah COCOMO pada model proses pengembangan non-sekuensial dan cepat, rekayasa ulang, pendekatan reuse, dan pendekatan berorientasi objek. COCOMO II memiliki tiga model yaitu aplikasi komposisi model, awal desain model dan model pos arsitektur (Malik et al., 2013). Berikut model COCOMO II yang dapat dilihat pada gambar 2.1.



Gambar 2. 1. Model COCOMO II

Keterangan gambar 2.1:

1. Aplikasi komposisi model adalah model yang digunakan untuk memperkirakan usaha dan jadwal pada proyek-proyek yang menggunakan *Aided Software Engineering Computer Integrated* yaitu alat untuk pengembangan aplikasi dengan cepat. Menggunakan poin objek untuk ukuran.
2. Awal desain model adalah model yang melibatkan eksplorasi perangkat lunak, sistem arsitektur, dan konsep operasi. Hal ini didasarkan pada titik fungsi, 7 faktor skala, dan 5 pengganda usaha.
3. Model pos arsitektur adalah model perpanjangan rinci dari model desain awal dan memperkirakan untuk seluruh siklus hidup pengembangan. Hal ini digunakan ketika desain selesai dan informasi rinci tentang proyek ini tersedia.

Dari tiga macam model pada gambar 2.1 tersebut, model yang berwarna abu-abu merupakan model pos arsitektur yang digunakan dalam penulisan ini. Yang mana untuk menghitung usaha atau upaya dinyatakan sebagai *Person Month* (PM). Hal ini menentukan upaya yang diperlukan untuk proyek berdasarkan ukuran software proyek di sumber baris ribuan kode (KSLOC) serta faktor biaya lainnya yang dikenal sebagai faktor skala dan usaha pengganda (C. H. S. Reddy & Raju, 2009). Kemampuan pemodelan baru yang utama pada COCOMO II adalah model *software sizing*, yang melibatkan *Object Points*, Fungsi Poin, dan sumber baris kode, model *nonlinear* untuk digunakan kembali pada perangkat lunak dan rekayasa ulang, pendekatan eksponen - *driver* untuk pemodelan disekonomi *software* skala relatif dan beberapa penambahan, penghapusan, dan *update cost driver* dari COCOMO sebelumnya. Persamaan pada COCOMO II, adalah sebagai berikut (Sharma, 2011).

Konstanta *Effort*:

$$A=2,94; B=0,91; \quad (1)$$

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^n EM_i \quad (2)$$

$$\text{dimana } E = B + 0,01 \times \sum_{j=1}^5 SF_j \quad (3)$$

Konstanta *Schedule*:

$$C=3,67; D=0,28 \quad (4)$$

$$TDEV_{NS} = C \times (PM_{NS})^F \quad (5)$$

$$\begin{aligned} \text{dimana } F &= D + 0,2 \times 0,01 \times \sum_{j=1}^5 SF_j \\ &= D + 0,2 \times (E - B) \end{aligned} \quad (6)$$

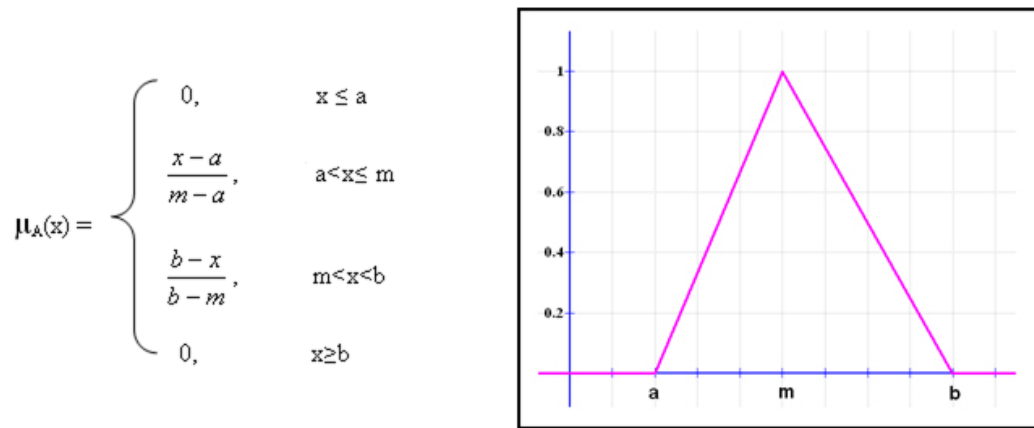
2.5. Fuzzy Gaussian

Logika *fuzzy* adalah metodologi sukses untuk memecahkan masalah realistik yang meliputi kesederhanaan dan fleksibilitas. Menangani masalah dengan data yang tidak tepat dan tidak lengkap yang didasarkan pada teori himpunan *fuzzy* yang diresmikan oleh Prof. Lofti Zadeh pada tahun 1965 (Stimulation, 2013). Hal ini digunakan untuk mewakili nilai-nilai linguistik seperti rendah, sedang dan kompleks. Set *fuzzy* direpresentasikan sebagai mana x adalah elemen di X dan $\mu_A(x)$ adalah fungsi keanggotaan himpunan A yang mendefinisikan keanggotaan himpunan *fuzzy* A dalam X (Stimulation, 2013). Sebuah himpunan *fuzzy* ditandai

dengan fungsi keanggotaan, yang berasosiasi dengan setiap titik dalam himpunan *fuzzy* sejumlah nyata dalam interval [0, 1], yang disebut derajat atau kelas keanggotaan. Fungsi keanggotaan terdiri dari *Triangular*, *Trapezoid*, *Gaussian* dan lain-lain (C. S. Reddy & Raju, 2009a).

2.5.1. Triangular Membership Function

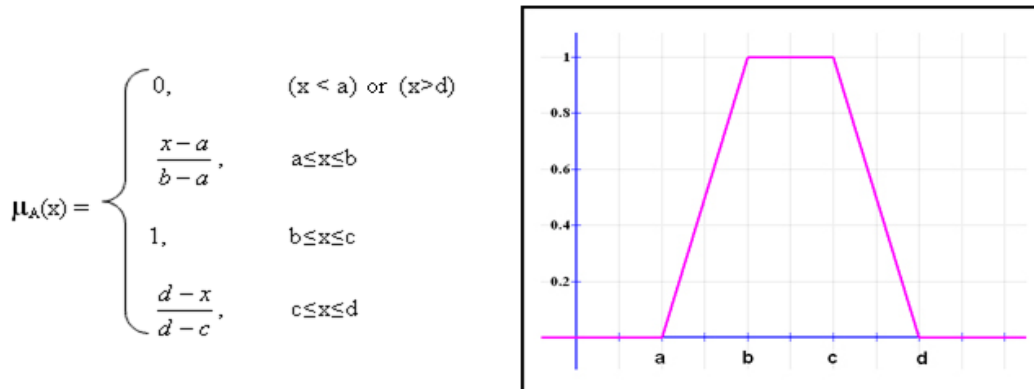
Sebuah fungsi *triangular* $\mu_A(x)$ yang bergantung pada parameter skala a sebagai batas bawah, b sebagai batas atas dan $a < m < b$. Gambar 2.2 akan menampilkan diagram *triangular*.



Gambar 2. 2. Triangular Membership Function

2.5.2. Trapezoidal Membership Function

Sebuah fungsi *trapezoidal* $\mu_A(x)$ yang dipengaruhi parameter scalar a sebagai batas bawah, d sebagai batas atas, b sebagai batas tengah bawah, c sebagai batas tengah atas, dan $a < b < c < d$. Gambar 2.3 akan menampilkan diagram dari *trapezoidal*.

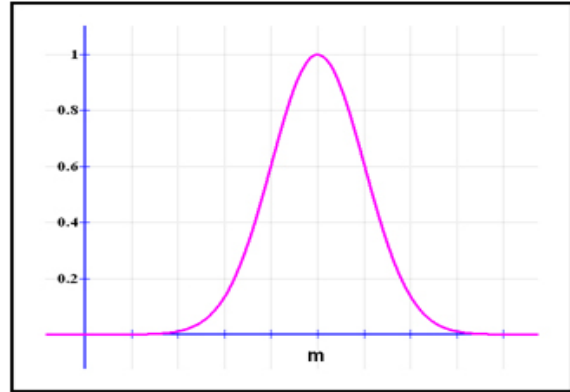


Gambar 2. 3. Trapezoidal Membership Function

2.5.3. Gaussian Membership Function

Sebuah fungsi *Gaussian* $\mu_A(x)$ yang bergantung pada nilai tengah m dan dengan sebuah standar deviasi $k > 0$. Semakin kecil nilai k maka semakin rapat kurva. Gambar 2.4 menampilkan kurva dari *Gaussian*.

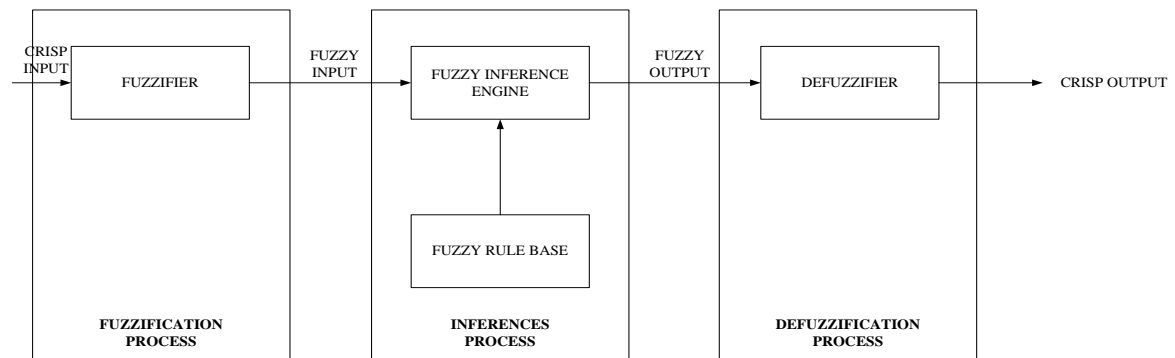
$$\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}}$$



Gambar 2. 4. Gaussian Membership Function

Fungsi keanggotaan *Gaussian* memberikan akurasi yang lebih baik daripada fungsi keanggotaan lainnya. *Gaussian* memiliki transisi yang lebih halus antar interval. Metode ini mengurangi penyimpangan antara estimasi dan biaya atau usaha sebenarnya (Stimation, 2013). Seperangkat aturan sudah ada batasnya untuk pemisah *fuzzy* mana yang menghasilkan upaya *software* sesuai dengan atribut nilai *fuzzy*. Upaya *software* ini terkelompok ke aturan yang beragam berdasarkan parameter dan teknik yang memanfaatkan prosedur optimasi dengan maksud untuk mengoptimalkan aturan *fuzzy* (G. Marappagounder, 2015).

Fuzzy Logic System (FLS) adalah nama yang diberikan untuk setiap system yang memiliki hubungan langsung dengan konsep *Fuzzy* dan *Fuzzy Logic*. *Fuzzy Logic System* yang terkenal dalam literatur dapat diklasifikasikan menjadi tiga jenis yaitu murni sistem logika *Fuzzy*, sistem *Fuzzy* sugeno, serta sistem logika *Fuzzy* dengan *fuzzifier* dan *defuzzifier*. Gambar 2.5 menunjukkan *Fuzzy Logic System* dengan *fuzzifier* dan *defuzzifier*.



Gambar 2. 5. Fuzzy Logic System dengan Fuzzifier dan Defuzzifier

- *Fuzzification process*: sebuah *fuzzifier* merubah *crisp input* menjadi nilai *fuzzy set* berdasarkan *membership function*. *Membership function* adalah kurva yang memetakan input menjadi *membership value* antara 0 dan 1.
- *Inference process*: menghasilkan *output*, berasal dari *fuzzy input* menggunakan *fuzzy rule*. *Fuzzy rule* IF menggunakan deskripsi *fuzzy* untuk mengukur input dan THEN adalah kesimpulan, menentukan semua kemungkinan output dari setiap input.
- *Defuzzification process*: proses, memproduksi dan menerjemahkan *fuzzy output* dari *interference process* menjadi kuantitatif. *Defuzzification* mengubah *fuzzy value* menjadi *crisp output*.

2.6. Bee Colony Optimization

Model COCOMO digunakan *software cost estimation* dan diperlukan waktu untuk perkembangan *system*. Karena meningkatnya *volume* dan kompleksitas proyek perangkat lunak maka metode tersebut kurang akurat. Maka digunakanlah model non-algoritma seperti model pembelajaran (Gharehchopogh & Dizaji, 2014). Dalam pendekatan analogi *fuzzy*, data kategoris dan numerik ditandai dengan *fuzzy set* dan karenanya menghasilkan aturan yang menggunakan penyebaran optimasi untuk mengoptimalkan aturan dalam *fuzzy* (G. Marappagounder, 2015). Sejumlah besar metode estimasi tersedia tetapi hanya sedikit yang dapat mencapai tingkat kepuasan. Karena kompleksitas dan tingginya persyaratan perangkat lunak menuntut hasil akurasi yang lebih baik, maka COCOMO II menjadi kurang tepat. Maka penelitian ini mengusulkan menggunakan *Bee Colony Optimization* (BCO) untuk memperbaiki akurasi dan mengoptimalkan biaya perangkat lunak (Chalotra

et al., 2015). *Bee Colony Optimization* (BCO) adalah contoh pendekatan sempurna untuk pemodelan jenis kecerdasan. BCO melakukan dengan pendekatan *bottom up* yang diciptakan dari analogi lebah. BCO digunakan sebagai solusi pencarian terbaik untuk masalah optimasi (Chalotra et al., 2015). Adapun rumus untuk menghitung *loyalty* dari *Bee Colony Optimization* (BCO) adalah sebagai berikut:

$$loyalty = \frac{max_sol_value - current_sol_value}{max_sol_value - min_sol_value} \quad (7)$$

Dimana, *max_sol_value* adalah nilai maksimum dari sekumpulan solusi, *min_sol_value* adalah nilai minimum dari sekumpulan solusi dan *current_sol_value* adalah nilai solusi saat ini.

[Halaman ini sengaja dikosongkan]

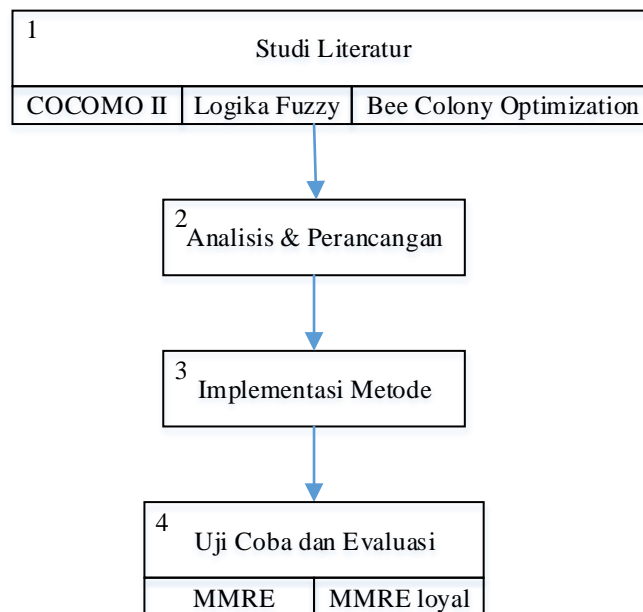
BAB 3

METODOLOGI PENELITIAN

Dalam bab ini akan dijelaskan tahapan penelitian, data input, rancangan metode, skenario uji coba, analisis hasil, dan evaluasi metode yang akan digunakan.

3.1. Tahapan Penelitian

Tahapan-tahapan yang dilakukan pada penelitian ini antara lain adalah studi literatur, analisis masalah, rancangan metode, implementasi metode, pengujian dan evaluasi. Secara lebih detail, penelitian ini dirancang dengan urutan tahapan seperti pada Gambar 3.1.



Gambar 3. 1. Tahap Penelitian

3.2. Studi Literatur

Studi literatur merupakan tahap untuk mempelajari konsep, teori, fakta dan informasi yang diperlukan dalam penelitian tesis ini. Pada suatu penelitian selalu diawali dengan proses pengkajian pustaka yang terkait dengan topik penelitian yang diambil. Dalam penelitian ini, referensi yang digunakan adalah literatur-literatur ilmiah yang berkaitan dengan perkiraan usaha dan biaya proyek perangkat lunak serta metode untuk meningkatkan akurasi dari estimasi. Hasil dari studi literatur adalah nilai rata-rata kesalahan pada estimasi usaha dan biaya perangkat lunak. Dari

hasil tersebut kemudian dilakukan analisa keterhubungan dengan topik terkait. Literatur-literatur yang berhubungan dengan topik penelitian akan dipelajari lebih lanjut. Literatur tersebut antara lain adalah sebagai berikut.

- Literatur tentang perkiraan usaha dan biaya proyek perangkat lunak.
- Literatur tentang *cost driver* COCOMO II yang akan digunakan di setiap metode dalam tesis ini.
- Literatur tentang metode *Fuzzy Gaussian* untuk meningkatkan akurasi estimasi.
- Literatur tentang metode *Bee Colony Optimization* untuk menghasilkan akurasi estimasi yang optimal.

3.3. Analisis dan Perancangan

3.3.1 Analisis

Permasalahan utama yang diangkat dalam tesis ini adalah bagaimana cara untuk mengoptimasi perkiraan biaya dari sebuah perangkat lunak dengan menggunakan COCOMO II *Fuzzy Gaussian*. Optimasi pada COCOMO II *Fuzzy Gaussian* dapat memberikan hasil yang lebih akurat dalam perkiraan biaya. Perkiraan yang tepat akan memberikan manajemen proyek perangkat lunak yang baik. Karena ketepatan dalam memperkirakan usaha dan biaya proyek perangkat lunak merupakan salah satu dari tiga aspek utama hubungan di dalam proyek.

Untuk mengimplementasikan metode ini digunakan dataset yang berasal dari *Turkish Software Industry* (Dataset, n.d.) dan Nasa (Nasa, n.d.). Untuk data yang berasal dari *Turkish Software Industry*, disebut dataset Cocomo-sdr. Data tersebut terdiri dari 12 proyek dan dari 5 perusahaan perangkat lunak yang berbeda. Sedangkan data yang berasal dari Nasa, disebut *dataset* Nasa 93-dem. Data tersebut juga terdiri dari 93 proyek. Masing-masing dataset memiliki 25 atribut, yaitu 17 *cost driver* COCOMO II, 5 *scale factor*, ID *project*, *actual effort* dan SLOC. *Dataset* tersebut kemudian dihitung menggunakan COCOMO II *Fuzzy Gaussian* lalu dibandingkan hasilnya dengan nilai *actual effort* yang terdapat pada data tersebut. Hasil dari perbandingan adalah nilai kesalahan atau MMRE pada estimasi COCOMO II *Fuzzy Gaussian*. Sehingga untuk memperkecil nilai kesalahan dan meningkatkan akurasi dari COCOMO II *Fuzzy Gaussian*, digunakan *Bee Colony*

Optimization. Input untuk menggunakan *Bee Colony Optimization* adalah nilai baru pada *effort multiplier* dari *Fuzzy Gaussian*. Sedangkan *output* dari *Bee Colony Optimization* pada masalah ini adalah MMRE loyal. *Cost driver* COCOMO II yang digunakan pada proyek perangkat lunak yang dikembangkan secara jelas dan pasti pada tahap awal dari siklus hidupnya, mempengaruhi hasil akurasi dari perkiraan usaha dan biaya proyek. Biasanya akan muncul bias ketika melakukan pengukuran *cost driver* yang menjadi penyebabnya adalah karena didasarkan pada penilaian manusia. Adapun nilai dari *effort multiplier* dari *cost driver* juga nilai dari *scale factor* COCOMO II dapat dilihat pada tabel 3.1 dan tabel 3.1 dibawah ini.

Tabel 3.1 Nilai Effort Multiplier COCOMO II

No	Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
1	RELY	0,82	0,92	1	1,1	1,26	-
2	DATA	-	0,9	1	1,14	1,28	-
3	RESU	-	0,95	1	1,07	1,15	1,24
4	DOCU	0,81	0,91	1	1,11	1,23	-
5	TIME	-	-	1	1,11	1,29	1,63
6	STOR	-	-	1	1,05	1,17	1,46
7	PVOL	-	0,87	1	1,15	1,3	-
8	ACAP	1,42	1,19	1	0,85	0,71	-
9	PCAP	1,34	1,15	1	0,88	0,76	-
10	PCON	1,29	1,12	1	0,9	0,81	-
11	APEX	1,22	1,1	1	0,88	0,81	-
12	PLEX	1,19	1,09	1	0,91	0,85	-
13	LTEX	1,2	1,09	1	0,91	0,84	-
14	TOOL	1,17	1,09	1	0,9	0,78	-
15	SITE	1,22	1,09	1	0,93	0,86	0,8
16	SCED	1,43	1,14	1	1	1	-
17	CPLX	0,73	0,87	1	1,17	1,34	1,74

Tabel 3.2 Nilai Scale Factor COCOMO II

SF	Very Low	Low	Nominal	High	Very High	Extra High
PREC	6,2	4,96	3,72	2,48	1,24	0
FLEX	5,07	4,05	3,04	2,03	1,01	0
RESL	7,07	5,65	4,24	2,83	1,41	0
TEAM	5,48	4,38	3,29	2,19	1,1	0
PMAT	7,8	6,24	4,68	3,12	1,56	0

Adapun penjelasan dari tabel 3.1 diatas adalah:

- RELY

Required Software Reliability, yaitu tingkatan *reliability* dari perangkat lunak yang akan dibangun. Tabel 3.3 menunjukkan deskripsi dari RELY.

Tabel 3.3 Deskripsi RELY

RELY Descriptor	Slight inconvenience	Low, easily recoverable losses	Moderate, easily recoverable losses	High financial loss	Risk to human life	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.82	0.92	1.00	1.10	1.26	n/a

Contoh penggunaan EM yang terdapat pada tabel 3.3 yaitu, jika proyek perangkat lunak akan dikembangkan untuk mengatur penerbangan pesawat maka *Required Software Reliability* (RELY) harus memiliki nilai 1,26. Karena pengaturan penerbangan pesawat berkaitan dengan kehidupan manusia. RELY berfungsi untuk mengukur performa *software* terhadap kegagalan. Adapun rentang nilai pada level RELY berkisar dari paling rendah yaitu 0,82 hingga sangat tinggi yaitu 1,26 dan tidak memiliki nilai ekstra tinggi. Berdasarkan contoh kasus tersebut dengan nilai RELY yang dicapai maka, efek dari kegagalan *software* sangat rendah dan tidak berisiko terhadap kehidupan manusia. Berdasarkan rentang nilai yang ada, jika RELY memiliki nilai 0,85 maka nilai tersebut berada pada level sangat rendah.

– DATA

Data *Base Size*, yaitu besaran data yang disimpan pada penyimpanan. Tabel 3.4 menunjukkan deskripsi dari DATA.

Tabel 3.4 Deskripsi DATA

RELY Descriptor	Slight inconvenience	Low, easily recoverable losses	Moderate, easily recoverable losses	High financial loss	Risk to human life	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.92	1.00	1.10	1.26	n/a

– RESU

Develop of Reusability, yaitu memperkirakan usaha yang dibutuhkan dalam rangka mengembangkan komponen yang akan digunakan kembali dalam proyek. Tabel 3.5 menunjukkan deskripsi dari RESU.

Tabel 3.5 Deskripsi RESU

RESU Descriptor		None	Across Project	Across program	Across product line	Across multiple product lines
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.95	1.00	1.07	1.15	1.24

– DOCU

Documentation Match to Life Cycle Needs, yaitu penjelasan mengenai kesesuaian dokumentasi proyek dengan kebutuhan. Tabel 3.6 pengembang perangkat lunak menunjukkan deskripsi dari DOCU.

Tabel 3.6 Deskripsi DOCU

DOCU Descriptor	Many life cycle needs uncovered	Some life cycle needs uncovered	Right sized to life cycle needs	Excessive for life cycle needs	Very excessive for life cycle needs	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.81	0.91	1.00	1.11	1.23	n/a

– TIME

Execution Time Constraint, menjelaskan presentasi ketepatan waktu eksekusi perintah yang diharapkan. Tabel 3.7 menunjukkan deskripsi dari TIME.

Tabel 3.7 Deskripsi TIME

TIME Descriptor			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.11	1.29	1.63

– STOR

Main Storage Constraint, yaitu merepresentasikan presentasi besaran penyimpanan yang digunakan. Tabel 3.8 menunjukkan deskripsi dari STOR.

Tabel 3.8 Deskripsi STOR

STOR Descriptor			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.00	1.17	1.48

– PVOL

Platform Volatility, yaitu menjelaskan kompleksitas perangkat lunak dan keras yang digunakan seperti OS dan DBMS. Tabel 3.9 menunjukkan deskripsi dari PVOL.

Tabel 3.9 Deskripsi PVOL

PVOL Descriptor		Major change every 12 mo., Minor change every 1 mo.	Major: 6 mo Minor: 2wk	Major:2 mo Minor: 1 wk	Major:2 wk Minor: 2 days	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.87	1.00	1.15	1.30	n/a

– ACAP

Analyst Capability, yaitu parameter yang menjelaskan kemampuan dari analis dalam mengobservasi kebutuhan serta desain. Tabel 3.10 menunjukkan deskripsi dari ACAP.

Tabel 3.10 Deskripsi ACAP

ACAP Descriptor	15 th percentile	35 th percentile	55 th percentile	75 percentile	90 th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.42	1.19	1.00	0.85	0.71	n/a

– PCAP

Programmer Capability, yaitu parameter yang menjelaskan kemampuan programmer sebagai sebuah tim yang mampu bekerjasama. Tabel 3.11 menunjukkan deskripsi dari PCAP.

Tabel 3.11 Deskripsi PCAP

PCAP Descriptor	15 th percentile	35 th percentile	55 th percentile	75 th percentile	90 th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.34	1.15	1.00	0.88	0.76	n/a

– PCON

Personnel Continuity, yaitu parameter yang mendeskripsikan persentase pergantian pegawai dalam setahun. Tabel 3.12 menunjukkan deskripsi dari PCON.

Tabel 3.12 Deskripsi PCON

PCON Descriptor	15 th percentile	35 th percentile	55 th percentile	75 th percentile	90 th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.42	1.19	1.00	0.85	0.71	n/a

– APEX

Application Experience, yaitu parameter yang menjelaskan tingkat pengalaman tim pengembang dalam mengembangkan perangkat lunak. Tabel 3.13 menunjukkan deskripsi dari APEX.

Tabel 3.13 Deskripsi APEX

APEX Descriptor	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.10	1.00	0.88	0.81	n/a

– PLEX

Platform Experience, yaitu parameter yang menjelaskan tingkat pengalaman tim pengembang dalam mengembangkan antar muka, baris data, jaringan dan distribusi sistem. Tabel 3.14 menunjukkan deskripsi dari PLEX.

Tabel 3.14 Deskripsi PLEX

PLEX Descriptor	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.19	1.09	1.00	0.91	0.85	n/a

– LTEX

Language and Tool Experience, yaitu parameter yang menjelaskan tingkat pengalaman tim pengembang dalam bahasa pemrograman dan penggunaan alat. Tabel 3.15 menunjukkan deskripsi dari LTEX.

Tabel 3.15 Deskripsi LTEX

LTEX Descriptor	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.20	1.09	1.00	0.91	0.84	

– TOOL

Use of Software Tools, yaitu parameter yang menjelaskan kompleksitas alat yang digunakan dalam pengembangan. Tabel 3.16 menunjukkan deskripsi dari TOOL.

Tabel 3.16 Deskripsi TOOL

TOOL Descriptor	Edit, code, debug	Sample frontend, backend CASE, little integration	Basic lifecycle tools, moderately integrated	Strong, mature lifecycle tools, moderately integrated	Strong, mature, proactive lifecycle tools, well integrated with processes methods, reuse	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.17	1.09	1.00	0.91	0.76	n/a

– SITE

Multisite Development, yaitu parameter yang merepresentasikan dua hal berupa distribusi data dan penunjang komunikasi. Tabel 3.17 menunjukkan deskripsi dari SITE.

Tabel 3.17 Deskripsi SITE

SITE: collocation descriptors:	International	Multicity or multi company	Same city or metro area	Same building or complex	Fully collocated	
SITE: communications descriptors:	Some phone, mail	Individual phone, fax	Narrow band email	Wideband electronic communication	Wideband electronic communication... occasional video conf.	Interactive multimedia
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.09	1.00	0.93	0.86	0.80

– SCED

Required Development Schedule, yaitu parameter yang menjelaskan persentase dari akselerasi penyelesaian proyek terhadap waktu yang diberikan. Tabel 3.18 menunjukkan deskripsi dari SCED.

Tabel 3.18 Deskripsi SCED

SCED Descriptor	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.43	1.14	1.00	1.00	1.00	n/a

– CPLX

Product Complexity, yaitu parameter yang dibagi dalam lima wilayah yaitu operasi pengendalian, komputasional, kemandirian perangkat, manajemen data, dan manajemen antarmuka. Tabel 3.19 menunjukkan deskripsi dari CPLX.

Tabel 3.19 Deskripsi CPLX

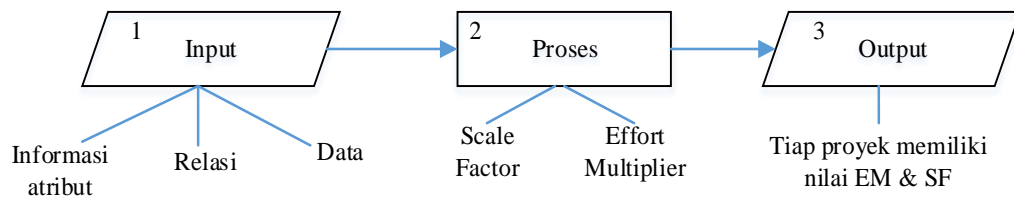
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.20	1.09	1.00	0.91	0.84	

Berikut ini adalah keterangan dari tabel 3.2 yaitu deskripsi 5 scale factor:

1. PREC. *Precendentedness*, yaitu parameter yang menjelaskan pengalaman terdahulu terhadap proyek yang identik dengan proyek yang sedang dikerjakan.
2. FLEX. *Development Flexibility*, yaitu parameter yang menjelaskan fleksibilitas dari proses pengembangan.
3. RESL. *Risk Resolution*, yaitu parameter yang menjelaskan sejauh mana analisis terhadap resiko.
4. TEAM. *Team Cohesion*, yaitu parameter yang menjelaskan sejauh mana anggota tim saling mengenal sesama anggota dalam tim.
5. PMAT. *Process Maturity*, yaitu parameter yang menjelaskan keseluruhan tingkat *maturity* dari perangkat lunak.

3.3.2 Perancangan Cost Driver COCOMO II

Setelah melakukan studi literatur dan pemahaman pada COCOMO II didapatkan, maka selanjutnya *dataset* yang ada diimplementasikan ke dalam bentuk hasil berupa nilai estimasi *effort* yaitu nilai estimasi yang nantinya digunakan untuk menghitung estimasi biaya dan juga usaha dari proyek perangkat lunak. Selanjutnya dihitung perbandingan nilai estimasi dengan nilai yang sebenarnya. Jika ternyata kurang akurat maka diimplementasikan metode untuk meningkatkan akurasi dari COCOMO II. Dataset yang diperoleh dari *Turkish Software Industry* dan Nasa menggunakan *cost driver* COCOMO II yang mana memiliki atribut terdiri dari *project ID*, *scale factor*, *effort multiplier* dan LOC. *Cost driver* dari COCOMO II tersebut diterbitkan oleh Barry Boehm pada tahun 1994 hasil pengembangan dari COCOMO sebelumnya yang terdiri dari 63 titik data dan seluruh titik data dibagi menjadi 16 variabel. COCOMO II dibuat berdasarkan proyek-proyek yang telah ada, seperti dari Nasa (Engineering, 2000). Adapun *cost driver* yang digunakan tersebut akan melalui beberapa tahapan terlebih dahulu sebelum digunakan oleh kedua *dataset*. Karena *cost driver* dan nilai pada masing-masing level *effort multiplier* dan *scale factor* yang didapatkan dari hasil penelitian Boehm masih terpisah dari tiap *dataset* yang akan digunakan. Berikut gambar 3.2 merupakan rancangan dari proses *cost driver* COCOMO II yang digunakan kedua *dataset* tersebut untuk estimasi usaha dan biaya proyek perangkat lunak.



Gambar 3. 2. Rancangan Cost Driver COCOMO II

Berikut penjelasan dari setiap tahapan rancangan pada gambar 3.2.

1. Input

Tahap pertama di dalam proses untuk menggunakan *cost driver* COCOMO II adalah mengambil informasi atribut yang terdiri dari 3 bagian yaitu deskripsi atribut, relasi atau hubungan antara atribut dan level, dan data. Adapun ketiga bagian tersebut dapat dilihat seperti berikut. Pada gambar 3.3 merupakan deskripsi atribut dari *cost driver* COCOMO II. Deskripsi tersebut digunakan untuk mempermudah dalam memahami maksud dari atribut-atribut yang ada di dalam COCOMO II.

```

- Exponential Cost Drivers:
  PREC: Precedentedness
  FLEX: Development Flexibility
  RESL: Arch/Risk Resolution
  TEAM: Team Cohesion
  PMAT: Process Maturity

- Product Factors
  RELY: Required Software Reliability
  DATA: Database Size
  CPLX: Product Complexity
  RUSE: Required Reusability
  DOCU: Documentation match to life-cycle needs

- Platform Factors
  TIME: Execution Time Constraint
  STOR: Main Storage Constraint
  PVOL: Platform Volatility

- Personnel Factors
  ACAP: Analyst Capability
  PCAP: Programmer Capability
  PCON: Personnel Continuity
  AEXP: Applications Experience
  PEXP: Platform Experience
  LTEX: Language and Tool Experience

- Project Factors
  TOOL: Use of Soft. Tools
  SITE: Multisite Development
  SCED: Development Schedule

- Input:
  LOC: Lines of Code

- Output:
  ACTUAL EFFORT: Effort spent for project in terms of man month

```

Gambar 3. 3. Deskripsi Atribut Cost Driver COCOMO II

```

@relation cocomo_sdr

@attribute ProjectID numeric
@attribute PREC {'Very High','Extra High',Nominal,'Very Low',High,Low}
@attribute FLEX {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute RESL {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute TEAM {High,'Very High','Extra High',Nominal,'Very Low',Low}
@attribute PMAT {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute RELY {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute DATA {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute CPLX {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute RUSE {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute DOCU {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute TIME {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute STOR {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute PVOL {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute ACAP {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute PCAP {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute PCON {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute AEXP {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute PEXP {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute LTEX {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute TOOL {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute SITE {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute SCED {Low,'Very Low',High,'Extra High','Very High',Nominal}
@attribute LOC numeric
@attribute 'ACTUAL EFFORT' numeric

```

Gambar 3. 4. Relasi Atribut Dengan Level di Cost Driver COCOMO II

```

@data

1,'Very High',Nominal,Low,High,'Very Low',Nominal,Nominal,Low,Low,Low,'Very High',High,Nominal,High,'Very High','Very High','Very High',High,High,High,Low,Nominal,3000,1,2

2,'Very High',Nominal,Low,'Very High','Very Low',High,Nominal,Low,Nominal,Low,'Very High',High,Nominal,High,High,'Very High','Very High','Very High','Very High',High,'Very Low',High,2000,2

3,'Extra High',Nominal,Low,High,'Very Low',High,'Very High',Low,Nominal,Low,'Very High','Very High',Nominal,High,High,'Very High','Very High','Very High','Very High',High,'Very Low',High,4250,4,5

4,'Extra High',High,High,High,Nominal,High,High,'Very High',Low,Nominal,Nominal,Nominal,High,High,High,'Very Low',High,'Very High',High,Nominal,High,High,10000,3

5,Nominal,Nominal,Low,High,Nominal,Low,Nominal,'Very High',High,Nominal,High,High,Nominal,Nominal,High,High,High,Nominal,Nominal,Nominal,High,Low,15000,4

6,'Very High',Low,'Extra High','Extra High',High,Low,Low,Nominal,Low,'Very Low',Nominal,Nominal,Low,'Very High','Very High',High,Nominal,High,High,'Very Low','Extra High',Low,40530,22

7,'Extra High','Very High','Extra High',Nominal,High,Low,Low,Low,Nominal,'Very Low',Nominal,Nominal,Low,'Very High','Very High','Very High',Nominal,'Very High','Very High','Very Low',High,Low,4050,2

8,Nominal,'Very Low','Very High',Nominal,Low,Low,'Very High','Very Low',High,'Very High','Very High','Very High',Low,Low,High,'Very High',Nominal,Nominal,Nominal,Nominal,'Very High',Nominal,31845,5

9,'Very Low','Very Low',Nominal,'Very Low',Low,'Very Low',Nominal,Low,'Very High',Low,Nominal,Nominal,Low,Low,High,'Very High',High,High,High,Nominal,High,Nominal,114280,18

10,Nominal,'Very Low','Very High',Low,Low,'Very Low',High,Nominal,'Very High','Very Low','Very High','Very High',Low,Low,'Very High','Very High',High,'Very High',High,Nominal,Nominal,Low,23106,4

11,High,Nominal,'Very High',High,Low,Low,Nominal,Nominal,Nominal,Low,Nominal,Nominal,Low,'Very High',High,'Very High',Nominal,Nominal,High,'Very High','Very High',High,1369,1

12,Low,Nominal,Nominal,High,Nominal,Low,Low,High,High,Nominal,Nominal,Nominal,Low,Low,High,'Very High',High,Nominal,High,'Very High','Very High',High,1611,2,1]

```

Gambar 3. 5. Level di Tiap Proyek

Pada gambar 3.4 merupakan tampilan relasi atribut dengan level yang ada pada *cost driver* COCOMO II. Relasi tersebut digunakan untuk mengetahui seperti apa level tiap proyek di tiap *cost driver*. Dari level tersebut dapat di perkirakan bagaimana usaha dan biaya yang dihasilkan dari proyek perangkat lunak.

Selain itu sebagai contoh pada gambar 3.5 di tahap ini menggunakan *dataset* Cocomo-sdr terdiri dari 12 proyek yang menampilkan isi dari level yang terdapat di dalam tiap proyek pada setiap *cost driver*. Sehingga dapat diketahui bagaimana level tiap *cost driver* di masing-masing proyek.

2. Proses

Tahapan ini menjelaskan tentang proses penggabungan antara level-level di tiap *scale factor* dan *effort multiplier* pada masing-masing proyek dengan nilai-nilai pada masing-masing *scale factor* dan *effort multiplier*. Yang mana tujuan dari proses tersebut dilakukan agar memudahkan dalam menentukan nilai pada tiap proyek di masing-masing *cost driver*. Sebagai contoh tampilan level-level di dalam *scale factor* dapat dilihat pada gambar 3.6.

ID	SCALE FACTOR				
	PREC	FLEX	RESL	TEAM	PMAT
1	Very High	Nominal	Low	High	Very Low
2	Very High	Nominal	Low	Very High	Very Low
3	Extra High	Nominal	Low	High	Very Low
4	Extra High	High	High	High	Nominal
5	Nominal	Nominal	Low	High	Nominal
6	Very High	Low	Extra High	Extra High	High
7	Extra High	Very High	Extra High	Nominal	High
8	Nominal	Very Low	Very High	Nominal	Low
9	Very Low	Very Low	Nominal	Very Low	Low
10	Nominal	Very Low	Very High	Low	Low
11	High	Nominal	Very High	High	Low
12	Low	Nominal	Nominal	High	Nominal

Gambar 3. 6. Level Scale Factor COCOMO II

Berikut nilai yang terdapat di dalam *scale factor* COCOMO II yang akan digabungkan dengan level-level pada gambar 3.6 diatas. Dalam hal menggabungkan antara level dan nilai tersebut akan tetap disesuaikan dengan jumlah proyek yang ada, yang mana disebutkan sebelumnya bahwa sebagai contoh

proyek yang digunakan pada tahap ini adalah proyek Cocomo-sdr. Adapun nilai *scale factor* dapat dilihat pada gambar 3.7 berikut.

SF	Very Low	Low	Nominal	High	Very High	Extra High
PREC	6,2	4,96	3,72	2,48	1,24	0
FLEX	5,07	4,05	3,04	2,03	1,01	0
RESL	7,07	5,65	4,24	2,83	1,41	0
TEAM	5,48	4,38	3,29	2,19	1,1	0
PMAT	7,8	6,24	4,68	3,12	1,56	0

Gambar 3. 7. Nilai Scale Factor COCOMO II

3. Output

Tahapan ini menjelaskan tentang hasil penggabungan antara level-level di tiap *scale factor* dan *effort multiplier* pada masing-masing proyek dengan nilai-nilai pada masing-masing *scale factor* dan *effort multiplier*. Adapun pada tahap ini mengambil contoh hasil dari penggabungan *scale factor*. Contoh penggabungan *scale factor* ini merupakan *cost driver* yang dapat digunakan untuk melakukan estimasi usaha dan biaya proyek perangkat lunak. Hasil dari penggabungan tersebut dapat dilihat pada gambar 3.8 berikut.

ID	PREC	FLEX	RESL	TEAM	PMAT
1	1,24	3,04	5,65	2,19	7,8
2	1,24	3,04	5,65	1,1	7,8
3	0	3,04	5,65	2,19	7,8
4	0	2,03	1,41	2,19	4,68
5	3,72	3,04	5,65	2,19	4,68
6	1,24	4,05	0	0	3,12
7	0	1,01	0	3,29	3,12
8	3,72	5,07	1,41	3,29	6,24
9	6,2	5,07	4,24	5,48	6,24
10	3,72	5,07	1,41	4,38	6,24
11	2,48	3,04	1,41	2,19	6,24
12	4,96	3,04	4,24	2,19	4,68

Gambar 3. 8. Hasil Penggabungan Level dan Nilai SF COCOMO II

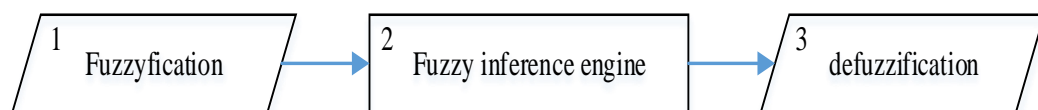
3.3.3 Memodelkan Fuzzy Logic Ke Dalam COCOMO II

Dalam usaha untuk meningkatkan akurasi COCOMO II, model COCOMO II yang sebelumnya sudah dibuat dimodelkan ke dalam *Fuzzy logic* yang bertujuan untuk membuat desain baru dari nilai *effort multiplier*. Cara optimasi nilai tersebut

dengan memodelkan *effort multiplier* ke dalam *fuzzy system*. *Membership function* pada *Fuzzy* yang digunakan untuk memodelkan adalah *Gaussian membership function* pada penelitian sebelumnya telah dilakukan perbandingan *membership function* yang hasilnya memberikan nilai yang lebih baik dibandingkan *membership function* lainnya seperti *triangular* dan *trapezoidal*.

Dengan mempelajari sifat pada *cost driver* COCOMO II, dapat dipahami bahwa untuk meningkatkan akurasi dapat menggunakan *Gaussian membership function* untuk merepresentasikan *cost driver*. *Input* pada *fuzzy* digunakan *descriptor* pada *effort multiplier*. Serta sebagai *output* digunakan nilai *effort multiplier* itu sendiri. Nilai *effort multiplier* dapat diubah ke dalam *fuzzy set*. Namun tidak semua *effort multiplier* dapat diimplementasikan *fuzzy*. *Effort multiplier* yang bersifat kualitatif adalah seperti RELY dimana nilai *high* apabila perangkat lunak itu dapat mengancam kehidupan manusia. Sedangkan yang bersifat kuantitatif adalah seperti LTEX, dimana nilai nominal adalah jika tim pengembang perangkat lunak telah berpengalaman menggunakan *tool* untuk pengembangan selama satu tahun. Nilai tersebut diukur tidak berdasarkan subyektifitas. Adapun 11 *effort multiplier* yang bersifat kuantitatif yang digunakan pada *fuzzy Gaussian* yaitu: DATA, STOR, TIME, ACAP, PCAP, APEX, PLEX, LTEX, SCED, PVOL, PCON.

Adapun tahapan *fuzzy logic* yang memproses *effort multiplier* COCOMO II menjadi *effort multiplier* kuantitatif dengan nilai yang baru, dapat dilihat pada gambar 3.9 berikut.

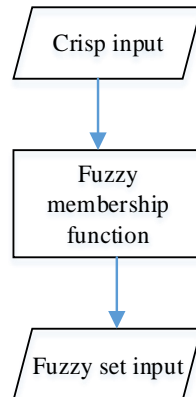


Gambar 3. 9. Tahapan Implementasi Fuzzy

Setiap tahapan rancangan pada gambar 3.9 menghasilkan nilai baru pada *effort multiplier* yang bersifat kuantitatif. *Effort multiplier* dengan nilai yang baru tersebut diharapkan dapat menurunkan kesalahan estimasi yang dihasilkan oleh COCOMO II, sehingga dengan begitu akurasi dapat ditingkatkan. Secara lengkap maksud dari tahapan implementasi *fuzzy* adalah sebagai berikut:

1. Fuzzyfication

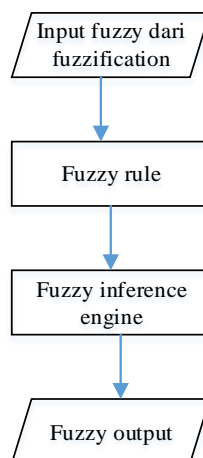
Langkah awal dalam implementasi *fuzzy* adalah *fuzzification*. *Fuzzification* adalah proses mengubah input berupa *crisp* data ke dalam *fuzzy set*. Sebelas *effort multiplier* akan dijadikan input *fuzzy set* menggunakan *Gaussian membership function* pada proses *fuzzification*. Berikut ini adalah diagram alir dari proses *fuzzification* seperti yang dapat dilihat pada gambar 3.10.



Gambar 3. 10. Proses Fuzzification

2. Fuzzy Inference Engine

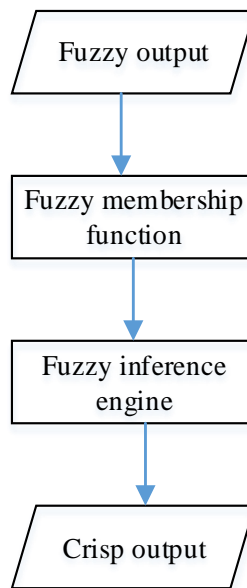
Fuzzy Inference Engine berguna untuk membuat *fuzzy rules* pada *fuzzy set*. *Fuzzy rule* berisi variabel linguistik atau verbal yang berhubungan dengan proyek. *Fuzzy rule* berfungsi untuk memetakan antara *input* dan *output* pada *fuzzy set*. Jumlah *rule* pada *fuzzy set* bergantung pada jumlah rentang yang ada pada *descriptor effort multiplier*. Berikut ini adalah diagram alir dari proses *fuzzy inference engine* seperti yang dapat dilihat pada gambar 3.11.



Gambar 3. 11. Proses pada Fuzzy Inference Engine

3. Deffuzification

Langkah terakhir pada *fuzzy* adalah *defuzzification* yaitu mengubah *fuzzy output* menjadi *crisp output*. Hasil dari *fuzzy input* yang kemudian diberikan *rule* maka akan menjadi *fuzzy output*, *fuzzification* mengubah *fuzzy output* menjadi *crisp output* yang kemudian menghasilkan nilai baru dari *effort multiplier*. Berikut ini adalah diagram alir dari proses *deffuzification* seperti yang dapat dilihat pada gambar 3.12.



Gambar 3. 12. Proses pada deffuzification

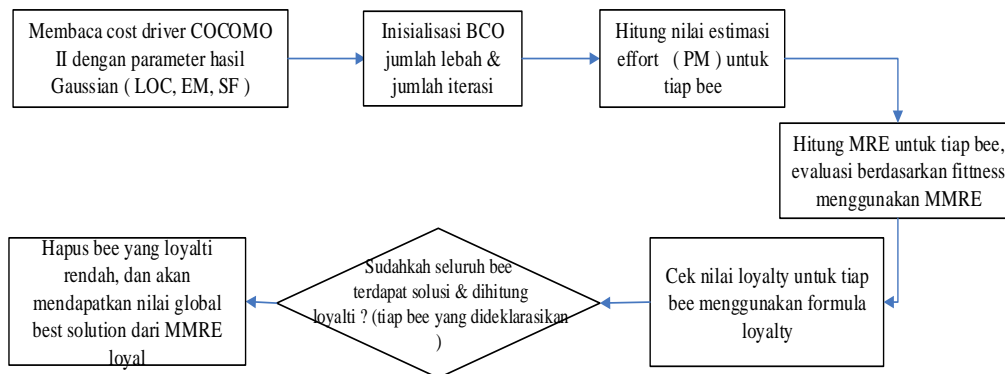
3.3.4 Perancangan COCOMO II Fuzzy Gaussian Dengan BCO

Metode *Bee Colony Optimization* digunakan untuk meningkatkan akurasi estimasi dari metode COCOMO II *Fuzzy Gaussian* dan agar mendapatkan hasil yang optimal. Nilai pada parameter A dan B di COCOMO II diubah dengan cara menaikkan dan menurunkan nilai parameter tersebut. Penaikan dan penurunan nilai pada parameter A dan B di dasarkan pada gradual awal yaitu 0,01 hingga gradual tertentu.

3.4 Implementasi Metode

Pada tahap implementasi, alur yang dilakukan pada tesis ini adalah seperti pada gambar 3.13. Tahap implementasi dimulai dari memasukkan *cost driver*, LOC, SF dari COCOMO II dan nilai EM dari *Fuzzy Gaussian*. Kemudian di proses

dengan cara menentukan jumlah lebah dan iterasi. Dan juga menghitung nilai MRE dari tiap lebah. Hingga dihasilkan nilai optimal yaitu MMRE loyal. MMRE loyal dipilih yang paling minimum nilainya diantara MMRE loyal dari iterasi yang lainnya. Alur proses implementasi BCO dapat dilihat pada gambar 3.13 dibawah ini.



Gambar 3. 13. Alur Proses Implementasi BCO

Berikut deskripsi dari gambar 3.13 yaitu alur proses implementasi dari *Bee Colony Optimization*, yang mana inputnya berupa *cost driver* COCOMO II dan nilai *effort multiplier* hasil dari implementasi *Fuzzy Gaussian* dan outputnya berupa MMRE loyal yang paling minimum.

- Menerima *input* data dari *dataset* Cocomo-sdr dan Nasa93-dem.
- Memastikan jumlah lebah, jumlah iterasi dan jumlah pembagian kelompok lebah menjadi 3 bagian.
- Melakukan inisialisasi parameter A dan parameter B, LOC, SF dari COCOMO II dan EM dari *Fuzzy Gaussian*.
- Menyiapkan modul BCO agar bisa dipanggil.
- Menghitung nilai *effort* (E) dari tiap lebah.
- Menghitung MRE di tiap lebah yang sudah mencari nilai E (*Effort*).
- Menghitung MMRE dari keseluruhan MRE, jika hasil MMRE mendekati *actual effort* (AE) maka dipilih.
- Menghitung nilai ‘loyalti’ dari setiap lebah yang nantinya akan digunakan sebagai faktor proses seleksi.

- Mengulang langkah ke-5 hingga langkah ke-8 untuk setiap lebah.
- Menghapus solusi yang tidak loyal (optimal), yaitu solusi yang kurang dari *threshold* yang ditentukan.
- Menentukan nilai solusi yang paling mendekati nilai MMRE.
- Menghentikan proses BCO jika solusi terbaik ditemukan, yaitu *best solution* dengan nilai MMRE loyal paling minimum.

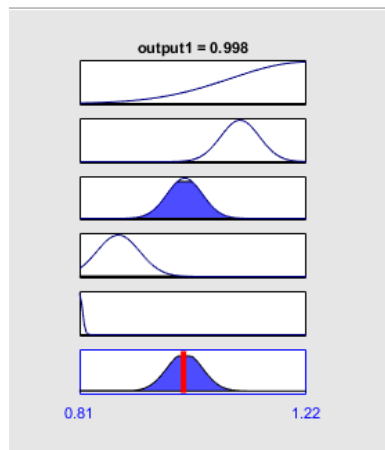
3.5. Uji Coba dan Evaluasi

3.5.1 Uji Coba

Setelah tahapan implementasi metode selesai, maka tahapan selanjutnya adalah melakukan suatu uji coba terhadap metode yang telah diusulkan. Uji coba dimaksudkan untuk mengetahui apakah penelitian yang dilakukan telah dapat memenuhi tujuan penelitian sebagaimana yang telah direncanakan. Pada tahap ini akan dilakukan uji coba untuk meningkatkan nilai akurasi dari estimasi usaha dan biaya proyek perangkat lunak menggunakan metode *Bee Colony Optimization* yang akan dibandingkan dengan metode COCOMO II *Fuzzy Gaussian*. Sebagai contoh menggunakan *dataset* Cocomo-sdr. Setelah memasukkan *cost driver* COCOMO II, adapun beberapa tahapan uji coba pada penulisan tesis ini diantaranya yaitu:

- Menghitung dan menampilkan dataset dengan *cost driver* COCOMO II dan nilai *effort multiplier* yang baru setelah dilakukan implementasi menggunakan *Fuzzy Gaussian*, seperti yang dapat di lihat pada gambar 3.14 dibawah ini.

Gambar 3.14 adalah hasil dari salah satu *effort multiplier* COCOMO II yaitu APEX yang diubah nilainya menjadi *effort multiplier* dengan nilai yang baru.



Gambar 3. 14. Nilai Effort Multiplier APEX

Gambar 3.14 menjelaskan bahwa nilai *effort multiplier* APEX yang berasal dari COCOMO II berada pada level 1,22; 1,1; 1; 0,88; hingga 0,81. Setelah melakukan implementasi pada *Fuzzy Gaussian*, nilai APEX pada level nominal diubah menjadi 0,998 yang mana sebelumnya adalah bernilai 1. Hal tersebut terjadi karena diambil nilai tengah antara rentang nilai *effort multiplier* tersebut.

- Menghasilkan perubahan nilai *effort multiplier* baru pada beberapa *cost driver* lainnya, yang ditandai dengan warna abu-abu seperti yang terlihat pada tabel 3.20.

Tabel 3.20 Effort Multiplier Fuzzy Gaussian

No.	Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
1	RELY	0,82	0,92	1	1,1	1,26	-
2	DATA	-	0,9	1	1,14	1,24	-
3	RESU	-	0,95	1	1,07	1,15	1,24
4	DOCU	0,81	0,91	1	1,11	1,23	-
5	TIME	-	-	1	1,11	1,29	1,59
6	STOR	-	-	1	1,05	1,16	1,45
7	PVOL	-	0,87	1	1,14	1,23	-
8	ACAP	1,41	1,18	1,01	0,85	0,74	-
9	PCAP	1,34	1,15	1	0,88	0,76	-
10	PCON	1,26	1,12	1	0,9	0,81	-
11	APEX	1,11	1,1	0,998	0,88	0,81	-
12	PLEX	1,13	1,06	1	0,91	0,857	-
13	LTEX	1,16	1,1	1	0,91	0,85	-
14	TOOL	1,17	1,09	1	0,9	0,78	-
15	SITE	1,22	1,09	1	0,93	0,86	0,8
16	SCED	1,31	1,14	1	1	1	-
17	CPLX	0,73	0,87	1	1,17	1,34	1,74

- Menghitung akurasi estimasi usaha dan biaya menggunakan *effort multiplier* yang baru dengan implementasi *Fuzzy Gaussian*. Hasil yang diperoleh adalah MMRE atau total *error* dari kesalahan atau MRE setiap proyek di dalam dataset. Proses dan hasil hitungan dengan implementasi *Fuzzy Gaussian* ada pada gambar 3.21.

ID	E	EM	A	B	C	D	PM	AE	MRE
1	1,1092	0,35077	2,94	0,91	3,67	0,28	3,48809	1,2	190,67%
2	1,0983	0,45924					2,89076	2	44,54%
3	1,0968	0,62912					9,04272	4,5	100,95%
4	1,0131	1,1116					33,6818	3	1022,73%
5	1,1028	1,08193					63,0292	4	1475,73%
6	0,9941	0,24742					28,8455	22	31,12%
7	0,9842	0,20684					2,40897	2	20,45%
8	1,1073	1,03007					139,808	5	2696,16%
9	1,1823	0,37025					295,104	18	1539,47%
10	1,1182	0,63899					62,916	4	1472,90%
11	1,0636	0,23405					0,96103	1	3,90%
12	1,1011	0,40746					2,02519	2,1	3,56%
							MMRE FUZZY		716,85%

Gambar 3. 15. Hitung Estimasi Dengan Implementasi Fuzzy Gaussian

- Memasukkan seluruh *cost driver* COCOMO II dengan *effort multiplier* yang baru dari *Fuzzy Gaussian* ke metode *Bee Colony Optimization*.
- Menghitung nilai akurasi dari estimasi usaha dan biaya menggunakan metode *Bee Colony Optimization*. Hasil yang paling optimal, yaitu MMRE loyal terdapat pada gradual 0,0001. Nilai MMRE loyal dibandingkan antar tiap iterasi dan nilai optimal yang digunakan adalah nilai MMRE loyal yang paling minimum diantara lainnya. Seperti pada contoh gambar 3.16 dibawah ini merupakan MMRE loyal dari *dataset* Cocomo-sdr dengan 10 iterasi.

1	2	3	4	5	6	7	8	9	10
8.0705	8.0459	8.0212	7.9966	7.9743	7.9789	7.9836	7.9883	7.9930	7.9977

Gambar 3. 16. MMRE loyal Hasil Implementasi Estimasi Dengan BCO

3.5.2 Evaluasi

Evaluasi pada pengujian di dalam penulisan ini adalah membandingkan hasil akurasi estimasi usaha dan biaya perangkat lunak menggunakan metode

COCOMO II *Fuzzy Gaussian* dan *Bee Colony Optimization* pada masing-masing *dataset*. Dimana dalam tiap metode tersebut akan dilakukan perhitungan untuk rata-rata nilai kesalahan. Rata-rata nilai tersebut di dapatkan dari total kesalahan tiap proyek di dalam *dataset*. Dan hasil akhir adalah memperoleh nilai akurasi yang optimal (MMRE loyal).

Untuk mendapatkan hasil akurasi yang optimal tersebut maka akan dianalisa menggunakan perubahan nilai pada parameter A dan parameter B dengan cara nilai pada parameter tersebut dinaikkan dan diturunkan berdasarkan gradual yang telah ditentukan. Yang mana gradual awal adalah 0,01. Hingga ditemukan nilai akurasi yang optimal pada gradual tertentu maka iterasi akan dihentikan. Adapun cara menghitung rata-rata nilai kesalahan (MMRE) pada metode COCOMO II *Fuzzy Gaussian* terdapat pada persamaan (8) sedangkan cara untuk menghitung loyaliti pada metode *Bee Colony Optimization* terdapat pada persamaan (9) berikut.

$$MMRE = \frac{1}{N} \sum_{x=1}^N MRE \quad (8)$$

Dimana nilai N merupakan jumlah banyaknya proyek dalam suatu *dataset*. Sedangkan MRE adalah nilai kesalahan dari tiap proyek di dalam *dataset*. Dibawah ini adalah persamaan (9) yang merupakan persamaan untuk mencari loyaliti. Dari nilai loyaliti ini lah akan di dapatkan MMRE loyal dengan nilai paling minimum diantara nilai MMRE loyal pada iterasi lainnya.

$$loyalty = \frac{\max_sol_value - current_sol_value}{\max_sol_value - \min_sol_value} \quad (9)$$

Dimana \max_sol_value adalah nilai maksimum dari sekumpulan solusi. Sedangkan \min_sol_value adalah nilai minimum dari sekumpulan solusi. Dan $current_sol_value$ adalah nilai solusi saat ini. Hasil perbandingan dari akurasi estimasi usaha dan biaya proyek perangkat lunak dibagi menjadi dua bagian. Yaitu MMRE dari kedua *dataset* dengan metode COCOMO II *Fuzzy Gaussian* dan MMRE loyal dari kedua *dataset* dengan metode *Bee Colony Optimization*.

BAB 4

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan tahap uji coba dan evaluasi dari penelitian yang telah dilakukan. Tahapan implementasi metode peningkatan akurasi estimasi usaha dan biaya perangkat lunak pada COCOMO II ini dilakukan sesuai dengan desain sistem pada Gambar 3.1. Tahapan yang terdapat pada bab 3 terdiri dari pengurangan nilai *error* menggunakan COCOMO II *Fuzzy Gaussian* yang dilanjutkan dengan mengoptimalkan hasil dari *Fuzzy Gaussian* menggunakan *Bee Colony Optimization*. Tahap berikutnya uji coba dan evaluasi hasil. Uji coba dilakukan sesuai dengan skenario pengujian yang telah dirancang pada Sub Bab 3.5. Pada bagian akhir bab ini adalah hasil peningkatan akurasi estimasi usaha dan biaya perangkat lunak serta mengoptimalkan hasil dari akurasi pada *dataset* dan data uji.

4.1. Implementasi Penelitian

Pada penelitian ini, tahapan implementasi dan pengujian dilakukan dengan menggunakan perangkat keras dan perangkat lunak yang digunakan untuk proses peningkatan akurasi estimasi dan biaya perangkat lunak pada COCOMO II *Fuzzy Gaussian* dan *Bee Colony Optimization*.

4.2. Perancangan Uji Coba

Pada sub bab ini menjelaskan tentang pembagian *dataset*, skenario uji coba, proses implementasi teknik pengujian estimasi usaha dan biaya proyek serta analisa dan evaluasi dari hasil pengujian yang dilakukan.

4.2.1. Pembagian Dataset

Dataset yang digunakan pada uji coba penelitian ini adalah *dataset Cocomo_sdr* dari *Turkish Software Industry* dan *Nasa 93-dem* dari Nasa. Penelitian ini menggunakan 17 *cost driver*, 5 *scale factor* (SF) dari COCOMO II. Setiap nilai menentukan hasil dari estimasi biaya perangkat lunak. Nilai tersebut menilai bagaimana perangkat lunak yang akan dibangun melalui estimasi biaya yang dihasilkan. Setiap nilai memiliki rentang antara *very low* hingga *extra high*. *Cost*

driver adalah faktor-faktor yang mempengaruhi biaya pengembangan proyek. Tujuh belas *cost driver* pada COCOMO II dikelompokkan ke dalam 4 bagian yaitu:

- *Product attributes* (RELY, DATA, CPLX, RESU, DOCU).
- *Platform attributes* (TIME, STOR, PVOL).
- *Personnel attributes* (ACAP, PCAP, PCON, AEXP, PEXP, LTEX).
- *Project attributes* (TOOL, SITE, SCED).

Semakin besar nilai STOR, DATA, TIME, TURN, VIRT, CPLX, dan RELY, semakin besar pula biaya estimasi yang dihasilkan. Sedangkan untuk nilai ACAP, PCAP, AEXP, MODP, TOOL, VEXP, LEXP yang semakin besar akan menurunkan biaya estimasi (Arinta, Yusuf, & Widyastutik, 2012). Terdapat parameter lain yang digunakan yaitu *scale factor* (PREC, FLEX, RESL, TEAM, PMAT).

Masing-masing *cost driver* mempunyai level mulai dari *very low*, *low*, *nominal*, *high*, *very high* maupun *extra high*. Setiap rating memiliki nilai bobot, dimana nilai yang lebih besar dari 1 menunjukkan usaha pengembangan yang meningkat, sedangkan nilai dibawah 1 menunjukkan menurun. Nilai bobot ini selanjutnya digunakan untuk mencari *Effort Adjustment Factor* (EAF) dengan cara mengalikan setiap nilai bobot dari *cost driver* yang digunakan dalam pengembangan proyek. Nilai EAF dan faktor skala akan dijadikan parameter dalam proses *effort equation* yaitu estimasi dari usaha yang dibutuhkan dalam menyelesaikan proyek berdasarkan ukuran proyek PM (dalam satuan orang-bulan). Selanjutnya COCOMO II akan memprediksi jumlah bulan yang dibutuhkan berdasarkan nilai *effort equation*. Kelebihan model ini dapat digunakan untuk memprediksi waktu yang dibutuhkan dalam menyelesaikan proyek dan jumlah staf yang dibutuhkan (Arinta et al., 2012).

4.3. Skenario Uji Coba

Berdasarkan metode yang diusulkan, penelitian ini memiliki beberapa tahapan proses. Tahap selanjutnya adalah melakukan implementasi. Implementasi yang dilakukan untuk melihat unjuk kerja yang dihasilkan oleh metode yang diusulkan. Data masukan pada uji coba ini adalah Cocomo-sdr dan Nasa 93-dem. Data yang digunakan diperoleh dari *Turkish Software Industry* terdiri atas 12 titik

data sedangkan Nasa 93-dem terdiri atas 93 titik data. Masing-masing terdiri dari 25 atribut, 17 *cost driver*, 5 *scale factor*, SLOC, *actual effort*. Terdapat dua skenario uji coba yang dilakukan pada penelitian ini, yaitu:

1. Implementasi hasil akurasi estimasi usaha dan biaya menggunakan COCOMO II *Fuzzy Gaussian* pada dataset Cocomo-sdr dan membandingkan hasil akurasi estimasi usaha dan biaya menggunakan COCOMO II *Fuzzy Gaussian* pada dataset Nasa 93-dem.
2. Implementasi hasil akurasi estimasi usaha dan biaya menggunakan *Bee Colony Optimization* pada dataset Cocomo-sdr dan membandingkan hasil akurasi estimasi usaha dan biaya menggunakan *Bee Colony Optimization* pada dataset Nasa 93-dem.

4.4. Implementasi COCOMO II Fuzzy Gaussian

Implementasi ini dilakukan dengan cara mengubah *effort multiplier* yang berupa *crisp* data ke dalam *fuzzy set* dengan *Gaussian membership function*. Untuk membuat estimasi biaya perangkat lunak dibutuhkan *input* dan *output* serta *rule* yang ditetapkan di dalam *Fuzzy*. Hasil dari *input* dan *output* serta *rule* tersebut akan diubah menjadi *crisp* data. Pada tesis ini menggunakan *Gaussian membership function*, maka *input* yang diperlukan berjumlah tiga titik. Tiap *cost driver* dan tingkatannya pada COCOMO II dimasukkan ke dalam implementasi *Fuzzy* guna menentukan rentang nilai *very low* hingga *extra high* pada *Fuzzy Gaussian*. Jika seluruh nilai *effort multiplier* pada COCOMO II telah dimasukkan ke dalam *Fuzzy Gaussian* sesuai *crisp* data pada *Fuzzy*, maka akan ditemukan perubahan nilai *effort multiplier* yang akan mempengaruhi nilai dari estimasi biaya. Sebagai contoh, dapat dilihat pada tabel 4.1 terdapat rentang nilai untuk TIME. Pada tabel tersebut rentang nilai yang dimiliki TIME dimulai dari nilai yang paling rendah 1,00 hingga nilai paling tinggi 1,63. Setiap puncak pada output disesuaikan dengan setiap nilai *effort multiplier*.

Tabel 4.1 TIME Descriptor

TIME Descriptor			≤ 50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.11	1.29	1.63

Pada tabel 4.1 memperlihatkan nilai *effort multiplier* tanpa perubahan yang berasal dari COCOMO II. Kemudian pada saat menggunakan *Fuzzy Gaussian*, nilai *effort multiplier* tersebut akan mengalami perubahan di beberapa nilai *cost driver*. Seperti nilai *effort multiplier Fuzzy Gaussian* pada tabel 3.20 yang akan digunakan untuk melakukan estimasi terhadap biaya perangkat lunak. Perubahan nilai *effort multiplier* yang diimplementasi menggunakan *Fuzzy Gaussian* ditandai dengan warna abu-abu.

Berikut contoh penggunaan *Effort Multiplier* dan *Scale Factor*, jika proyek perangkat lunak akan dikembangkan untuk mengatur penerbangan pesawat maka *Required Software Reliability* (RELY) harus memiliki nilai 1,26. Karena pengaturan penerbangan pesawat berkaitan dengan kehidupan manusia. RELY berfungsi untuk mengukur performa perangkat lunak terhadap kegagalan. Adapun rentang nilai pada level RELY berkisar dari paling rendah yaitu 0,82 hingga sangat tinggi yaitu 1,26 dan tidak memiliki nilai ekstra tinggi. Berdasarkan contoh kasus tersebut dengan nilai RELY yang dicapai maka, efek dari kegagalan perangkat lunak sangat rendah dan tidak berisiko terhadap kehidupan manusia. RELY berfungsi untuk mengukur performa perangkat lunak terhadap kegagalan. Berdasarkan rentang nilai yang ada, jika RELY memiliki nilai 0,85 maka nilai tersebut berada pada level sangat rendah.

4.5. Implementasi Bee Colony Optimization Pada COCOMO II

Sub bab ini akan menjelaskan bagaimana implementasi dari *Bee Colony Optimization* (BCO) pada COCOMO II *Fuzzy Gaussian*. Nilai baru dari *effort multiplier* yang dihasilkan dari implementasi *Fuzzy* digunakan untuk menghitung nilai pada perhitungan COCOMO II. Nilai tersebut yang digunakan untuk

implementasi pada BCO. Implementasi BCO dapat menggunakan pemrograman Matlab. Hasil implementasi adalah berupa nilai baru dari parameter pada perhitungan COCOMO II *Fuzzy Gaussian*.

Implementasi tersebut membutuhkan beberapa masukan seperti *actual effort*, SLOC, *effort multiplier*, *scale factor*. Perhitungan pada implementasi ini adalah dengan mencari solusi dari nilai parameter A dan parameter B untuk setiap kemungkinan yang ada. Pembuatan solusi pada setiap perulangan dilakukan dengan menaikkan dan menurunkan nilai parameter A dan parameter B sebesar 0,01. Pencarian dilakukan hingga memenuhi keadaan berhenti. Batasan pencarian juga dapat dilakukan berdasarkan jumlah perulangan yang akan dilakukan menggunakan BCO. Setiap kemungkinan akan dicari untuk mendapatkan nilai terbaik dari parameter perhitungan COCOMO II. BCO dimulai dengan menentukan nilai solusi awal, kemudian dilakukan pencarian untuk kemungkinan nilai solusi baru, kemudian dilakukan pengecekan apakah solusi baru memenuhi kriteria, dan jika telah memenuhi kondisi berhenti maka nilai solusi baru tersebut adalah yang menjadi nilai solusi akhir dari metode ini. Selanjutnya dijelaskan langkah yang dilakukan dalam BCO. Penentuan solusi dilakukan dengan menaikkan dan menurunkan nilai pada parameter. Pada tesis ini terdapat dua *dataset* yang menggunakan *cost driver* serta parameter A dan parameter B dari COCOMO II. Berikut tabel dan gambar grafik yang menampilkan perubahan nilai pada masing-masing parameter serta dilengkapi dengan gradual nilai parameter pada kedua *dataset* yang terdiri dari tiga tahapan perubahan gradual. Sebagai contoh hal tersebut dilakukan guna mencapai hasil MMRE yang optimal. Perubahan nilai parameter yang pertama dilakukan pada dataset Cocomo-sdr. Tahap awal dapat dilihat pada tabel 4.2 dengan gradual 0,01.

Tabel 4.2 Cocomo-sdr Dengan Gradual 0,01

Solusi parameter A	Solusi parameter B	MMRE loyal (%)
0,19	0,91	9,21

Berdasarkan tabel 4.2 solusi parameter A terdapat pada iterasi ke-19 sedangkan solusi parameter B terdapat pada iterasi ke-91. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 295 kali. Dan dicari nilai paling minimal

diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 0 hingga 2,94. Rentang nilai pada tiap iterasi parameter B adalah nilai 0 hingga 0,91. Tahap kedua dapat dilihat pada tabel 4.3 dengan gradual 0,001.

Tabel 4.3 Cocomo-sdr Dengan Gradual 0,001

Solusi parameter A	Solusi parameter B	MMRE loyal (%)
0,189	0,905	8,01

Berdasarkan tabel 4.3 solusi parameter A terdapat pada iterasi ke-1 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 0,185 hingga 0,195. Rentang nilai pada tiap iterasi parameter B adalah nilai 0,905 hingga 0,915. Tahap kedua dapat dilihat pada tabel 4.4 dengan gradual 0,0001.

Tabel 4.4 Cocomo-sdr Dengan Gradual 0,0001

Solusi parameter A	Solusi parameter B	MMRE loyal (%)
0,1894	0,9045	7,9743

Berdasarkan tabel 4.4 solusi parameter A terdapat pada iterasi ke-5 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah nilai 0,1889 hingga 0,1899. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0,9045 hingga 0,9055.

Perubahan nilai parameter yang pertama dilakukan pada dataset Nasa 93-dem. Tahap awal dapat dilihat pada tabel 4.5 dengan gradual 0,01.

Tabel 4.5 Nasa 93-dem Dengan Gradual 0,01

Solusi parameter A	Solusi parameter B	MMRE loyal (%)
2,95	0,91	28,01

Berdasarkan tabel 4.5 solusi parameter A terdapat pada iterasi ke-295 sedangkan solusi parameter B terdapat pada iterasi ke-91. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 295 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada iterasi parameter A adalah

antara nilai 0 hingga 2,94. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0 hingga 0,91. Tahap kedua dapat dilihat pada tabel 4.6 dengan gradual 0,001.

Tabel 4.6 Nasa 93-dem Dengan Gradual 0,001

Solusi parameter A	Solusi parameter B	MMRE loyal (%)
2,955	0,905	1,6927345

Berdasarkan tabel 4.6 solusi parameter A terdapat pada iterasi ke-10 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 2,945 hingga 2,955. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0,905 hingga 0,915. Tahap ketiga dapat dilihat pada tabel 4.7 dengan gradual 0,0001.

Tabel 4.7 Nasa 93-dem Dengan Gradual 0,0001

Solusi parameter A	Solusi parameter B	MMRE loyal (%)
2,9545	0,9045	12,92

Berdasarkan tabel 4.7 solusi parameter A terdapat pada iterasi ke-1 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 2,9545 hingga 2,9555. Rentang nilai pada tiap iterasi B adalah antara nilai 0,9045 hingga 0,9055.

4.6. Hasil Uji Coba Skenario 1

Uji coba pertama melakukan implementasi *Fuzzy* pada COCOMO II menggunakan metode *Fuzzy Gaussian* pada *dataset* Nasa 93-dem dan membandingkan dengan penelitian sebelumnya yang menggunakan *dataset* Cocomo-sdr. Beberapa *Effort Multiplier* diubah nilainya ketika menggunakan *Fuzzy Gaussian*, hal ini dilakukan untuk meminimalkan *error*. Uji coba pertama ini dimaksudkan untuk mengetahui hasil dari unjuk kerja metode yang menggunakan COCOMO II *Fuzzy Gaussian*. Tabel 4.8 menunjukkan hasil implementasi pada skenario uji coba pertama. Jika hasil MMRE dari *dataset* Cocomo-sdr tersebut di

implementasikan ke dalam bentuk biaya estimasi usaha perangkat lunak maka, uang senilai Rp 1.000.000,00 yang dimiliki akan ditambahkan dengan hasil MMRE yang ada yaitu 716,85%. Maka hasilnya Rp 8.168.500,00 yang artinya *actual* Rp 1.000.000,00 dan estimasinya adalah Rp 8.168.500,00.

Tabel 4.8 MMRE Dengan Metode Cocomo II Fuzzy Gaussian

Dataset	MMRE (%)
Cocomo-sdr	716,85
Nasa 93-dem	51,08

4.7. Hasil Uji Coba Skenario 2

Uji coba pertama melakukan implementasi *Bee Colony Optimization* dengan perubahan nilai parameter A dan parameter B dan gradual tertentu pada *dataset* Cocomo-sdr dan Nasa 93-dem serta membandingkan dengan penelitian sebelumnya. Beberapa *Effort Multiplier* diubah nilainya ketika menggunakan *Bee Colony Optimization*, hal ini dilakukan untuk meminimalkan *error*. Uji coba pertama ini dimaksudkan untuk mengetahui hasil dari unjuk kerja metode yang menggunakan *Bee Colony Optimization*. Tabel 4.9 menunjukkan hasil implementasi pada skenario uji coba pertama. Jika hasil MMRE dari *dataset* Nasa 93-dem tersebut di implementasikan ke dalam bentuk biaya estimasi usaha perangkat lunak maka, uang senilai Rp 1.000.000,00 yang dimiliki akan ditambahkan dengan hasil MMRE loyal yang ada yaitu 12,92%. Maka hasilnya Rp 1.129.200,00 yang artinya *actual* Rp 1.000.000,00 dan estimasinya adalah Rp 1.129.200,00.

Tabel 4.9 MMRE loyal Dengan Metode Bee Colony Optimization

Dataset	MMRE loyal (%)
Cocomo-sdr	7,9743
Nasa 93-dem	12,92

4.8. Analisis Hasil

Bagian ini menganalisis hasil uji coba yang telah dilakukan. Uji coba tersebut menggunakan dua skenario berbeda. *Dataset* yang digunakan adalah *dataset* Cocomo-sdr yang diperoleh dari *Turkish Software Industry* memiliki 12 proyek dan *dataset* Nasa 93-dem yang diperoleh dari Nasa. Pengujian awal menggunakan

COCOMO II *Fuzzy Gaussian*. Metode tersebut digunakan dengan alasan karena hasil estimasi dari COCOMO II belum dapat meningkatkan akurasi. Disebabkan oleh hasil *error* yang masin besar. Sehingga hasil yang diperoleh masih jauh dari nilai *Actual Effort*. Metode *Fuzzy Gaussian* dapat menurunkan *error* lebih baik daripada metode COCOMO II. Dikarenakan adanya perubahan *Effort Multiplier* yang berupa *crisp* data ke dalam *fuzzy set* dengan *Gaussian membership function*. Setelah hasil estimasi *Gaussian* diperoleh, ternyata hasil MMRE masih belum dapat mencapai akurasi yang mendekati *Actual Effort*. Penulisan ini mengusulkan metode *Bee Colony Optimization* untuk mengoptimasi hasil dari estimasi usaha dan biaya perangkat lunak dari metode COCOMO II *Fuzzy Gaussian*. *Bee Colony Optimization* membutuhkan beberapa masukan seperti *Actual Effort*, SLOC, *Effort Multiplier*, *Scale Factor*. Perhitungan pada implementasi adalah dengan mencari solusi dari nilai parameter A dan parameter B untuk setiap kemungkinan yang ada. Pembuatan solusi pada setiap perulangan dilakukan dengan menaikkan dan menurunkan kedua nilai parameter tersebut. Pencarian dilakukan hingga memenuhi keadaan berhenti. *Bee Colony Optimization* dimulai dengan menentukan nilai solusi awal kemudian dilakukan pencarian untuk kemungkinan nilai solusi baru memnuhi kriteria. Jika telah memenuhi kondisi berhenti maka nilai solusi baru tersebut menjadi nilai solusi akhir.

Pada penulisan ini terdapat dua *dataset* yang menggunakan *cost driver* serta parameter A dan parameter B dari COCOMO II. Terdapat beberapa tabel dan gambar grafik yang menampilkan perubahan nilai. Perubahan nilai pada masing-masing parameter serta dilengkapi dengan gradual nilai parameter pada kedua *dataset* yang terdiri dari 3 tahapan perubahan gradual. Hal tersebut dilakukan guna mencapai hasil MMRE yang optimal.

Berdasarkan tabel 4.2 solusi parameter A terdapat pada iterasi ke-19 sedangkan solusi parameter B terdapat pada iterasi ke-91. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 295 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 0 hingga 2,94. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0 hingga 0,91.

Berdasarkan tabel 4.3 solusi parameter A terdapat pada iterasi ke-1 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 0,185 hingga 0,195. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0,905 hingga 0,915.

Berdasarkan tabel 4.4 solusi parameter A terdapat pada iterasi ke-5 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 0,1889 hingga 0,1899. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0,9045 hingga 0,9055.

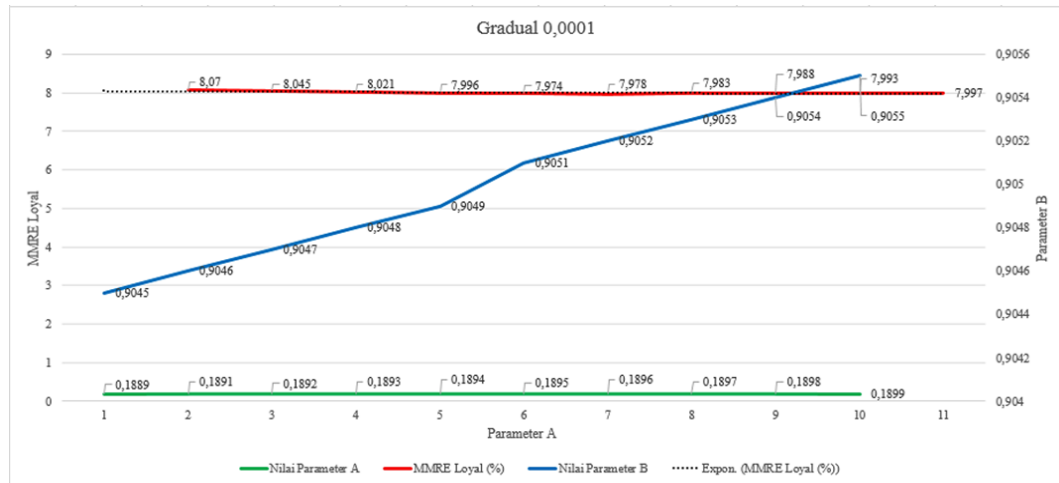
Berdasarkan tabel 4.5 solusi parameter A terdapat pada iterasi ke-295 sedangkan solusi parameter B terdapat pada iterasi ke-91. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 295 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 0 hingga 2,94. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0 hingga 0,91.

Berdasarkan tabel 4.6 solusi parameter A terdapat pada iterasi ke-10 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 2,945 hingga 2,955. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0,905 hingga 0,915.

Berdasarkan tabel 4.7 solusi parameter A terdapat pada iterasi ke-1 sedangkan solusi parameter B terdapat pada iterasi ke-1. Nilai tersebut didapatkan setelah melakukan iterasi sebanyak 10 kali. Dan dicari nilai paling minimal diantara seluruh nilai di tiap iterasi. Rentang nilai pada tiap iterasi parameter A adalah antara nilai 2,9545 hingga 2,9555. Rentang nilai pada tiap iterasi parameter B adalah antara nilai 0,9045 hingga 0,9055.

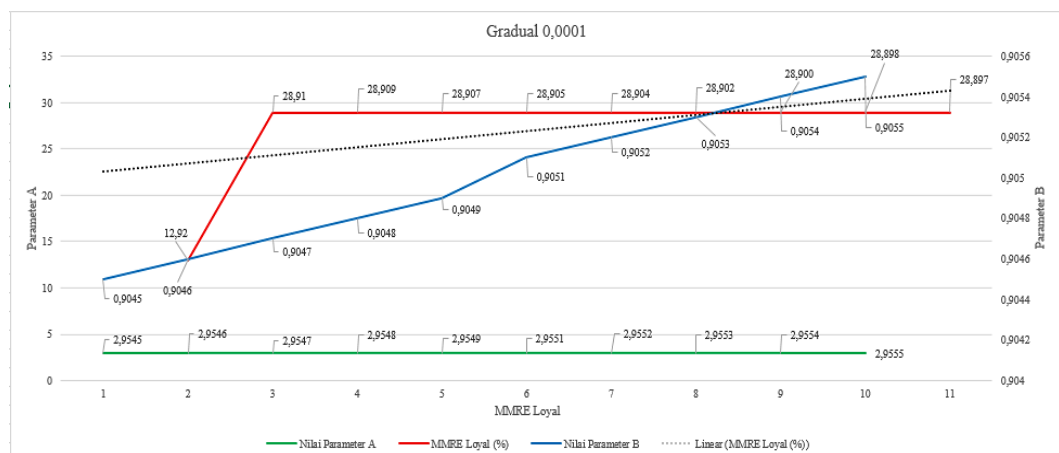
Hasil yang di analisa pada bagian ini adalah hasil akurasi MMRE dari kedua dataset dari pengujian tahap pertama hingga tahap ketiga menggunakan metode

BCO. Grafik yang menampilkan hasil solusi terbaik dari tiap uji coba dapat dilihat pada gambar 4.1 dan gambar 4.2 yang merupakan hasil akhir dari MMRE loyal dengan *error* paling minimal. Adapun gambar 4.1 dapat dilihat seperti dibawah ini.



Gambar 4. 1. Grafik MMRE dataset Cocomo-sdr

Pada gambar 4.1 menjelaskan MMRE loyal dengan gradual 0,0001 dari *dataset* Cocomo-sdr. Grafik menunjukkan *trendline* menurun, tetapi tidak secara signifikan. Dimana nilai paling rendah berada pada titik 7,9743. Hal tersebut dikarenakan rentang hasil nilai MMRE loyal tiap iterasi tidak terlalu besar. Sedangkan pada gambar 4.2 menjelaskan MMRE loyal dengan gradual 0,0001 dari *dataset* Nasa 93-dem.



Gambar 4. 2. Grafik MMRE loyal dataset Nasa 93-dem

Grafik pada gambar 4.2 menunjukkan *trendline* menaik tetapi tidak secara signifikan. Hal tersebut dikarenakan rentang hasil nilai MMRE loyal tiap iterasi tidak terlalu besar. Nilai MMRE loyal paling rendah berada pada titik 12,92. Dari hasil tersebut, solusi dengan gradual 0,0001 merupakan solusi terbaik dibandingkan solusi lain karena memiliki hasil MMRE loyal yang paling minimum serta akurasi estimasi yang optimal.

BAB 5

PENUTUP

Penelitian ini telah menjelaskan tentang metode COCOMO II *Fuzzy Gaussian* dan *Bee Colony Optimization* untuk meningkatkan akurasi usaha dan biaya proyek perangkat lunak. Perancangan, uji coba dan analisis hasil telah dilakukan pada penelitian ini. Pada bab ini akan dipaparkan kesimpulan dari penelitian yang telah dilakukan dan saran untuk pengembangan penelitian tentang akurasi estimasi usaha dan biaya proyek perangkat lunak.

5.1. Kesimpulan

Kesimpulan yang dapat diambil dalam penelitian ini antara lain adalah sebagai berikut.

- a. Berdasarkan hasil uji coba dan analisa hasil pengujian peningkatan akurasi estimasi usaha dan biaya proyek perangkat lunak pada metode COCOMO II *Fuzzy Gaussian* dan metode *Bee Colony Optimization* dengan *dataset* Cocomo-sdr dan *dataset* Nasa 93-dem, maka nilai parameter A dan parameter B diubah secara menaik dan menurun dari gradual 0,01 guna menurunkan *error* dalam estimasi usaha. Sedangkan hasil optimal yang didapatkan pada gradual 0,0001.
- b. Evaluasi yang dilakukan melalui MMRE merupakan nilai rata-rata dari MRE seluruh data. MME loyal menggunakan *Bee Colony Optimization* dapat menurunkan *error* lebih banyak daripada menggunakan COCOMO II *Fuzzy Gaussian*. Hal tersebut terbukti dari MMRE loyal yang dihasilkan menggunakan *dataset* Cocomo-sdr adalah 7,9743%. Sedangkan menggunakan *dataset* Nasa 93-dem adalah 12,92%. Maka dapat disimpulkan bahwa metode yang diusulkan menghasilkan nilai MMRE loyal lebih optimal dan dapat menurunkan *error* 38% lebih baik dibandingkan penelitian sebelumnya.
- c. Hasil MMRE loyal menggunakan *dataset* Nasa 93-dem lebih besar daripada jika menggunakan *dataset* Cocomo-sdr dengan metode usulan yang sama. Hal tersebut disebabkan oleh dua faktor, pertama karena MMRE loyal yang

dihasilkan pada *dataset* Nasa 93-dem dengan gradual 0,01 berada di iterasi ke-295. Dimana nilai parameter A yaitu 2,95. Sehingga nilai parameter A di setiap perubahan gradual dan di setiap iterasi mencapai 2,9555 pada gradual akhir yang optimal yaitu 0,0001. Kedua, hasil MMRE loyal yang dihasilkan pada *dataset* Cocomo-sdr dengan gradual 0,01 berada pada iterasi ke-19. Dimana nilai parameter A yaitu 0,19. Sehingga nilai parameter A di setiap perubahan gradual dan di setiap iterasi mencapai 0,1899 pada gradual akhir yang optimal yaitu 0,0001.

5.2. Saran

Untuk penelitian selanjutnya, disarankan memperbaiki hasil MMRE loyal pada penelitian ini. Karena, hasil MMRE loyal pada *dataset* Nasa 93-dem lebih besar daripada hasil MMRE loyal pada *dataset* Cocomo-sdr. Penelitian selanjutnya dapat menggunakan *dataset* berbeda seperti dataset kontekstual Indonesia atau jumlah *cost driver* yang berbeda. Dengan *dataset* atau jumlah *cost driver* yang berbeda maka, akurasi estimasi akan semakin meningkat karena jumlah data *training* dan *testing* akan semakin banyak dan semakin banyak pula bagian dari proyek perangkat lunak yang dapat di estimasi.

DAFTAR PUSTAKA

- Arinta, R., Yusuf, D., & Widyastutik, D. (2012). Manajemen Risiko Pada Model Estimasi Biaya Perangkat Lunak. *JUTI*, 10, 46–51.
- Borade, J. G., & Khalkar, V. R. (2013). Software Project Effort and Cost Estimation Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(8), 730–739. Retrieved from www.ijarcse.com
- Chalotra, S., Sehra, S. K., Brar, Y. S., & Kaur, N. (2015). Tuning of COCOMO Model Parameters by using Bee Colony Optimization. *Indian Journal of Science and Technology*, 8(July).
<https://doi.org/10.17485/ijst/2015/v8i14/70010>
- Dataset, P. (n.d.). cocomo-sdr. Retrieved September 10, 2016, from <https://terapromise.csc.ncsu.edu/repo/effort/cocomo/cocomo2/cocomo-sdr/cocomo-sdr.arff>
- Eberendu, A. C. (2014). Software Project Cost Estimation : Issues , Problems and Possible Solutions. *International Journal of Engineering Science Invention*, 3(6), 38–43. Retrieved from [http://www.ijesi.org/papers/Vol\(3\)6/Version-1/E0361038043.pdf](http://www.ijesi.org/papers/Vol(3)6/Version-1/E0361038043.pdf)
- Engineering, S. (2000). *COCOMO II Model Definition Manual*.
- G. Marappagounder, T. K. (2015). An Efficient Software Cost Estimation Technique Using Fuzzy Logic With The Aid Of Optimization Algorithm. *International Journal of Innovative Computing, Information and Control*, 11, 587 – 597. Retrieved from <http://www.ijicic.org/ijicic-13-12007.pdf>
- Gharehchopogh, F. S., & Dizaji, Z. A. (2014). A New Approach in Software Cost Estimation with Hybrid of Bee Colony and Chaos Optimizations Algorithms. *MGNT Research Report*, 2(6), 1263–1271.
- Jaglan, V. (2016). Apply Fuzzy Optimization in Proficient Managing COCOMO Model Cost Drivers. *International Journal of Recent Aspects*, 3(1), 20–26.
- Khatibi, E. (2012). Efficient Indicators to Evaluate the Status of Software Development Effort Estimation inside the Organizations. *International Journal of Managing Information Technology*, 4(3), 23–32.
<https://doi.org/10.5121/ijmit.2012.4303>

- L.Dasari, M.A.Kavya, V.Jasti, M. A. (2015). An Analysis on the Role of Project Management in the Success Chronicle of Software Project. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(9), 91–94.
- Malik, A., Pandey, V., & Kaushik, A. (2013). An analysis of fuzzy approaches for COCOMO II. *International Journal of Intelligent Systems and Applications*, 5(5), 68–75. <https://doi.org/10.5815/ijisa.2013.05.08>
- My, K. T. Le. (2016). Applying Teaching-Learning To Artificial Bee Colony for Parameter Optimization of Software Effort. *Journal of Engineering Science and Technology*, (August).
- Nasa. (n.d.). nasa93-dem. Retrieved February 2, 2017, from <https://terapromise.csc.ncsu.edu/#!/repo/view/head/effort/cocomo/cocomo2/nasa93-dem/nasa93-dem.arff>
- Potdar, S. M., & Potdar, M. (2014). Factors Influencing on Cost Estimation For Software Development. *Global Journal of Advanced Engineering Technologies*, 3(2), 119–123.
- Reddy, C. H. S., & Raju, K. (2009). Improving the accuracy of effort estimation through Fuzzy set combination of size and cost drivers. *WSEAS Transactions on Computers*, 8(6), 926–936.
- Reddy, C. S., & Raju, K. (2009a). An Improved Fuzzy Approach for COCOMO 's Effort Estimation using Gaussian Membership Function. *Journal of Software*, 4(5), 452–459. Retrieved from <http://ojs.academypublisher.com/index.php/jsw/article/view/0405452459/1074>
- Reddy, C. S., & Raju, K. (2009b). Improving the accuracy of effort estimation through fuzzy set combination of size and cost drivers. *WSEAS Transactions on Computers*, 8(6), 926–936. Retrieved from <http://www.wseas.us/e-library/transactions/computers/2009/31-479.pdf>
- Savalgi, A. A., Jain, S., Andanappa, S. S., & Narayan, S. J. (2016). Software Effort Estimation using Fuzzy Logic Membership Functions. *International Journal of Computer Systems*, 03(05), 366–370.
- Sharma, T. N. (2011). Analysis of Software Cost Estimation using COCOMO II.

- International Journal of Science & Engineering Research*, 2(6), 1–5.
Retrieved from
[http://www.ijser.org/researchpaper/Analysis_of_Software_Cost_Estimation_](http://www.ijser.org/researchpaper/Analysis_of_Software_Cost_Estimation_using_COCOMO_II.pdf)
[using_COCOMO_II.pdf](http://www.ijser.org/researchpaper/Analysis_of_Software_Cost_Estimation_using_COCOMO_II.pdf)
- Stimulation, E. (2013). A Fuzzy Approach For Software Effort Estimation. *International Journal on Cybernetics & Informatics*, 2(1), 9–15. Retrieved from <http://airccse.org/journal/ijci/papers/2113ijci02.pdf>
- Waghmode, S., & Kolhe, K. (2014). A Novel Way of Cost Estimation in Software Project Development Based on Clustering Techniques. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(4), 3892–3899. Retrieved from http://www.ijircce.com/upload/2014/april/17L_ANovel.pdf

[Halaman ini sengaja dikosongkan]

LAMPIRAN

LAMPIRAN 1 Dataset Cocomo-sdr Tanpa Metode Fuzzy Gaussian

1. Level di Dalam SF dan EM Cost Driver COCOMO II

SCALE FACTOR						EFFORT MULTIPLIER											
ID	PREC	FLEX	RESL	TEAM	PMAT	RELY	DATA	CPLX	RUSE	DOCU	TIME	STOR	PVOL	ACAP	PCAP	PCON	APEX
1	Very High	Nominal	Low	High	Very Low	Nominal	Nominal	Low	Low	Low	Very High	High	Nominal	High	Very High	Very High	Very High
2	Very High	Nominal	Low	Very High	Very Low	High	Nominal	Low	Nominal	Low	Very High	High	Nominal	High	High	Very High	Very High
3	Extra High	Nominal	Low	High	Very Low	High	Very High	Low	Nominal	Low	Very High	Very High	Nominal	High	High	Very High	Very High
4	Extra High	High	High	High	Nominal	High	High	Very High	Low	Nominal	Nominal	Nominal	High	High	High	Very Low	High
5	Nominal	Nominal	Low	High	Nominal	Low	Nominal	Very High	Nominal	High	High	Nominal	Nominal	High	High	High	Nominal
6	Very High	Low	Extra High	Extra High	High	Low	Low	Nominal	Low	Very Low	Nominal	Nominal	Low	Very High	Very High	High	Nominal
7	Extra High	Very High	Extra High	Nominal	High	Low	Low	Low	Nominal	Very Low	Nominal	Nominal	Low	Very High	Very High	Very High	Nominal
8	Nominal	Very Low	Very High	Nominal	Low	Low	Very High	Very Low	High	Very High	Very High	Very High	Low	Low	High	Very High	Nominal
9	Very Low	Very Low	Nominal	Very Low	Low	Very Low	Nominal	Low	Very High	Low	Nominal	Nominal	Low	Low	High	Very High	High
10	Nominal	Very Low	Very High	Low	Low	Very Low	High	Nominal	Very High	Very Low	Very High	Very High	Low	Low	Very High	Very High	High
11	High	Nominal	Very High	High	Low	Low	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Low	Very High	High	Very High	Nominal
12	Low	Nominal	Nominal	High	Nominal	Low	Low	High	High	Nominal	Nominal	Nominal	Low	Low	High	Very High	High

PLEX	LTEX	TOOL	SITE	SCED	LOC	AE
High	High	High	Low	Nominal	3000	1.2
Very High	Very High	High	Very Low	High	2000	2
Very High	Very High	High	Very Low	High	4250	4.5
Very High	High	Nominal	High	High	10000	3
Nominal	Nominal	Nominal	High	Low	15000	4
High	High	Very Low	Extra High	Low	40530	22
Very High	Very High	Very Low	High	Low	4050	2
Nominal	Nominal	Nominal	Very High	Nominal	31845	5
High	High	Nominal	High	Nominal	114280	18
Very High	High	Nominal	Nominal	Low	23106	4
Nominal	High	Very High	Very High	High	1369	1
Nominal	High	Very High	Very High	High	1611	2.1

2. Nilai di Dalam SF dan EM Cost Driver COCOMO II

Effort Multiplier						
No.	Cost Driver	Very Low	Low	Nominal	High	Very High Extra High
1	RELY	0,82	0,92	1	1,1	1,26 -
2	DATA	-	0,9	1	1,14	1,28 -
3	RUSE	-	0,95	1	1,07	1,15 1,24
4	DOCU	0,81	0,91	1	1,11	1,23 -
5	TIME	-	-	1	1,11	1,29 1,63
6	STOR	-	-	1	1,05	1,17 1,46
7	PVOL	-	0,87	1	1,15	1,3 -
8	ACAP	1,42	1,19	1	0,85	0,71 -
9	PCAP	1,34	1,15	1	0,88	0,76 -
10	PCON	1,29	1,12	1	0,9	0,81 -
11	APEX	1,22	1,1	1	0,88	0,81 -
12	PLEX	1,19	1,09	1	0,91	0,85 -
13	LTEX	1,2	1,09	1	0,91	0,84 -
14	TOOL	1,17	1,09	1	0,9	0,78 -
15	SITE	1,22	1,09	1	0,93	0,86 0,8
16	SCED	1,43	1,14	1	1	1 -
17	CPLX	0,73	0,87	1	1,17	1,34 1,74

SF	Very Low	Low	Nominal	High	Very High	Extra High
PREC	6,2	4,96	3,72	2,48	1,24	0
FLEX	5,07	4,05	3,04	2,03	1,01	0
RESL	7,07	5,65	4,24	2,83	1,41	0
TEAM	5,48	4,38	3,29	2,19	1,1	0
PMAT	7,8	6,24	4,68	3,12	1,56	0

3. Dataset Cocomo-sdr dari Cost Driver COCOMO II Tanpa Metode Fuzzy Gaussian

ID	PREC	FLEX	RESL	TEAM	PMAT	RELY	DATA	CPLX	RUSE	DOCU	TIME	STOR	PVOL	ACAP	PCAP	PCON	APEX	PLEX	LTEX	TOOL	SITE	SCED	LOC	AE
1	1.24	3.04	5.65	2.19	7.8	1	1	0.87	0.95	0.91	1.29	1.05	1	0.85	0.76	0.81	0.81	0.91	0.91	0.9	1.09	1	3000	1.2
2	1.24	3.04	5.65	1.1	7.8	1.1	1	0.87	1	0.91	1.29	1.05	1	0.85	0.88	0.81	0.81	0.85	0.84	0.9	1.22	1	2000	2
3	0	3.04	5.65	2.19	7.8	1.1	1.28	0.87	1	0.91	1.29	1.17	1	0.85	0.88	0.81	0.81	0.85	0.84	0.9	1.22	1	4250	4.5
4	0	2.03	1.41	2.19	4.68	1.1	1.14	1.34	0.95	1	1	1	1.15	0.85	0.88	1.29	0.88	0.85	0.91	1	0.93	1	10000	3
5	3.72	3.04	5.65	2.19	4.68	0.92	1	1.34	1	1.11	1.11	1	1	0.85	0.88	0.9	1	1	1	1	0.93	1.14	15000	4
6	1.24	4.05	0	0	3.12	0.92	0.9	1	0.95	0.81	1	1	0.87	0.71	0.76	0.9	1	0.91	0.91	1.17	0.8	1.14	40530	22
7	0	1.01	0	3.29	3.12	0.92	0.9	0.87	1	0.81	1	1	0.87	0.71	0.76	0.81	1	0.85	0.84	1.17	0.93	1.14	4050	2
8	3.72	5.07	1.41	3.29	6.24	0.92	1.28	0.73	1.07	1.23	1.29	1.17	0.87	1.19	0.88	0.81	1	1	1	1	0.86	1	31845	5
9	6.2	5.07	4.24	5.48	6.24	0.82	1	0.87	1.15	0.91	1	1	0.87	1.19	0.88	0.81	0.88	0.91	0.91	1	0.93	1	114280	18
10	3.72	5.07	1.41	4.38	6.24	0.82	1.14	1	1.15	0.81	1.29	1.17	0.87	1.19	0.76	0.81	0.88	0.85	0.91	1	1	1.14	23106	4
11	2.48	3.04	1.41	2.19	6.24	0.92	1	1	1	0.91	1	1	0.87	0.71	0.88	0.81	1	1	0.91	0.78	0.86	1	1369	1
12	4.96	3.04	4.24	2.19	4.68	0.92	0.9	1.17	1.07	1	1	1	0.87	1.19	0.88	0.81	0.88	1	0.91	0.78	0.86	1	1611	2.1

4. Dataset Cocomo-sdr dari Cost Driver COCOMO II Fuzzy Gaussian

ID	PREC	FLEX	RESL	TEAM	PMAT	RELY	DATA	CPLX	RUSE	DOCU	TIME	STOR	PVOL	ACAP	PCAP	PCON	APEX	PLEX	LTEX	TOOL	SITE	SCED	LOC	AE
1	1.24	3.04	5.65	2.19	7.8	1	1	0.87	0.95	0.91	1.29	1.05	1	0.85	0.76	0.81	0.81	0.91	0.91	0.9	1.09	1	3000	1.2
2	1.24	3.04	5.65	1.1	7.8	1.1	1	0.87	1	0.91	1.29	1.05	1	0.85	0.88	0.81	0.81	0.85	0.85	0.9	1.22	1	2000	2
3	0	3.04	5.65	2.19	7.8	1.1	1.24	0.87	1	0.91	1.29	1.16	1	0.85	0.88	0.81	0.81	0.85	0.85	0.9	1.22	1	4250	4.5
4	0	2.03	1.41	2.19	4.68	1.1	1.14	1.34	0.95	1	1	1	1.14	0.85	0.88	1.29	0.88	0.85	0.91	1	0.93	1	10000	3
5	3.72	3.04	5.65	2.19	4.68	0.92	1	1.34	1	1.11	1.11	1	1	0.85	0.88	0.9	0.998	1	1	1	0.93	1.14	15000	4
6	1.24	4.05	0	0	3.12	0.92	0.9	1	0.95	0.81	1	1	0.87	0.74	0.76	0.9	0.998	0.91	0.91	1.17	0.8	1.14	40530	22
7	0	1.01	0	3.29	3.12	0.92	0.9	0.87	1	0.81	1	1	0.87	0.74	0.76	0.81	0.998	0.85	0.85	1.17	0.93	1.14	4050	2
8	3.72	5.07	1.41	3.29	6.24	0.92	1.24	0.73	1.07	1.23	1.29	1.16	0.87	1.18	0.88	0.81	0.998	1	1	1	0.86	1	31845	5
9	6.2	5.07	4.24	5.48	6.24	0.82	1	0.87	1.15	0.91	1	1	0.87	1.18	0.88	0.81	0.88	0.91	0.91	1	0.93	1	114280	18
10	3.72	5.07	1.41	4.38	6.24	0.82	1.14	1	1.15	0.81	1.29	1.16	0.87	1.18	0.76	0.81	0.88	0.85	0.91	1	1	1.14	23106	4
11	2.48	3.04	1.41	2.19	6.24	0.92	1	1	1	0.91	1	1	0.87	0.74	0.88	0.81	0.998	1	0.91	0.78	0.86	1	1369	1
12	4.96	3.04	4.24	2.19	4.68	0.92	0.9	1.17	1.07	1	1	1	0.87	1.18	0.88	0.81	0.88	1	0.91	0.78	0.86	1	1611	2.1

LAMPIRAN 2 Dataset Nasa 93-dem Tanpa Metode Fuzzy Gaussian

1. Level di Dalam SF dan EM Cost Driver COCOMO II

SCALE FACTOR										EFFORT MULTIPLIER														
ID	PREC	FLEX	RESL	TEAM	PMAT	RELY	DATA	GPLX	RUSE	DOCU	TIME	STOR	PTOL	ACAP	PCAP	PCOM	APEX	PLEX	LTEX	TOOL	SITE	SCED	LOC	AE
1	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	25900	117,6
2	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	24600	117,6
3	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	7700	31,2
4	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	8200	36
5	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	9700	25,2
6	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	2200	8,4
7	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	3500	10,8
8	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	66600	352,8
9	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Extra High	Extra High	Low	High	High	Nominal	High	Nominal	High	High	High	Nominal	7500	72
10	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	Very High	Nominal	Very High	Nominal	High	Nominal	Nominal	Nominal	20000	72
11	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	High	Nominal	Very High	Nominal	High	Nominal	Nominal	Nominal	6000	24
12	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	Very High	Nominal	Very High	Nominal	High	Nominal	Nominal	Nominal	100000	360
13	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	Nominal	Nominal	Very High	Nominal	Low	Nominal	Nominal	Nominal	11300	36
14	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	High	High	High	Nominal	High	Low	Very Low	Nominal	Nominal	Nominal	100000	215
15	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	High	Nominal	Very High	Nominal	High	Nominal	Nominal	Nominal	20000	48
16	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Very Low	Nominal	Nominal	Nominal	100000	360
17	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Extra High	Low	High	Very High	Nominal	Very High	Nominal	High	Nominal	Nominal	Nominal	150000	324
18	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	High	Nominal	Very High	Nominal	High	Nominal	Nominal	Nominal	31500	60
19	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Nominal	Low	High	High	Nominal	Very High	Nominal	High	Nominal	Nominal	Nominal	15000	48
20	High	High	High	Very High	Nominal	Nominal	Low	High	Nominal	Nominal	Nominal	Extra High	Low	High	Nominal	Nominal	High	Nominal	High	Nominal	Nominal	Nominal	32500	60
21	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Low	19700	60
22	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Nominal	66600	300
23	High	High	High	Very High	High	High	Low	High	Nominal	Nominal	Nominal	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Nominal	29500	120
24	High	High	High	Very High	Nominal	High	Nominal	Nominal	Nominal	Nominal	High	Nominal	Nominal	Nominal	High	Nominal	High	Nominal	Nominal	Nominal	Nominal	Nominal	15000	90
25	High	High	High	Very High	Nominal	High	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	High	Nominal	High	Nominal	Nominal	Nominal	Nominal	Nominal	38000	210

2. Dataset Nasa 93-dem dari Cost Driver COCOMO II Tanpa Fuzzy Gaussian

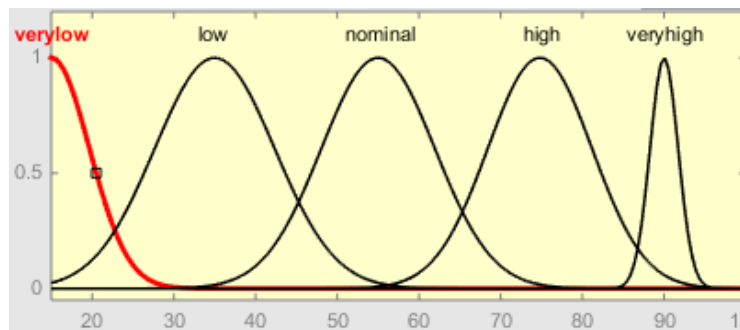
ID	PREC	FLEX	RESL	TEAM	PMAT	RELY	DATA	GPLX	RUSE	DOCU	TIME	STOR	PTOL	ACAP	PCAP	PCOM	APEX	PLEX	LTEX	TOOL	SITE	SCED	LOC	AE
1	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	25900	117,6
2	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	24600	117,6
3	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	7700	31,2
4	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	8200	36
5	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	9700	25,2
6	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	2200	8,4
7	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	3500	10,8
8	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	66600	352,8
9	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1,62	1,48	0,87	0,85	0,85	1	0,85	1	0,91	0,9	1	1	7500	72
10	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,85	1	0,81	1	0,91	1	1	1	20000	72
11	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,85	1	0,81	1	0,91	1	1	1	6000	24
12	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,76	1	0,81	1	0,91	1	1	1	100000	360
13	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	1	1	0,81	1	1,09	1	1	1	11300	36
14	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	1,15	0,85	0,85	1	0,85	1,09	1,2	1	1	1	100000	215
15	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,85	1	0,81	1	0,91	1	1	1	20000	48
16	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	1	1	1	1	1,2	1	1	1	100000	360
17	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1,48	0,87	0,85	0,85	1	0,81	1	0,91	1	1	1	150000	324
18	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,85	1	0,85	1	0,91	1	1	1	31500	60
19	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,85	1	0,81	1	0,91	1	1	1	15000	48
20	2,48	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1,48	0,87	0,85	1	1	0,85	1	0,91	1	1	1	32500	60
21	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	19700	60
22	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	66600	300
23	2,48	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1	1	1	1	1	0,91	1	1	1,14	29500	120
24	2,48	2,03	2,83	1,1	4,68	1,1	1	1	1	1	1,11	1	1	1	0,85	1	0,85	1	1	1	1	1	15000	90
25	2,48	2,03	2,83	1,1	4,68	1,1	1	1,17	1	1	1	1	1	1	0,85	1	0,85	1	1	1	1	1	38000	210

3. Dataset Nasa 93-dem dari Cost Driver COCOMO II Fuzzy Gaussian

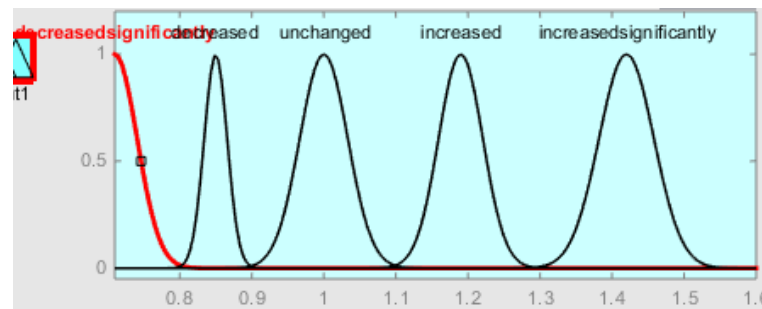
ID	PREC	FLEX	RESL	TEAM	PMAT	RELY	DATA	CPLX	RUSE	DOCU	TIME	STOR	PVOL	ACAP	PCAP	PCON	APEX	PLEX	LTEX	TOOL	SITE	SCED	LOC	AE
1	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	25900	117,6
2	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	24600	117,6
3	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	7700	31,2
4	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	8200	36
5	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	9700	25,2
6	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	2200	8,4
7	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	3500	10,8
8	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	66600	352,8
9	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1,59	1,45	0,87	0,85	0,88	1	0,88	1	0,91	0,9	1	1	7500	72
10	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,76	1	0,81	1	0,91	1	1	1	20000	72
11	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,88	1	0,81	1	0,91	1	1	1	6000	24
12	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,76	1	0,81	1	0,91	1	1	1	100000	360
13	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	1	1	0,81	1	1,1	1	1	1	11300	36
14	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	1,14	0,85	0,88	1	0,88	1,06	1,16	1	1	1	100000	215
15	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,88	1	0,81	1	0,91	1	1	1	20000	48
16	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	1	1	0,998	1	1,16	1	1	1	100000	360
17	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1,45	0,87	0,85	0,76	1	0,81	1	0,91	1	1	1	150000	324
18	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,88	1	0,88	1	0,91	1	1	1	31500	60
19	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1	0,87	0,85	0,88	1	0,81	1	0,91	1	1	1	15000	48
20	1,1	2,03	2,83	1,1	4,68	1	0,9	1,17	1	1	1	1,45	0,87	0,85	1	1	0,88	1	0,91	1	1	1	32500	60
21	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	19700	60
22	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	66600	300
23	1,1	2,03	2,83	1,1	3,12	1,1	0,9	1,17	1	1	1	1	0,87	1,01	1	1	0,998	1	0,91	1	1	1,14	29500	120
24	1,1	2,03	2,83	1,1	4,68	1,1	1	1	1	1	1,11	1	1	1,01	0,88	1	0,88	1	1	1	1	1	15000	90
25	1,1	2,03	2,83	1,1	4,68	1,1	1	1,17	1	1	1	1	1	1,01	0,88	1	0,88	1	1	1	1	1	36000	210

LAMPIRAN 3 Gaussian Membership Function

1. Input Cost Driver PCON Membership Function



2. Output Cost Driver PCON Membership Function



3. Rule Cost Driver PCON Membership Function

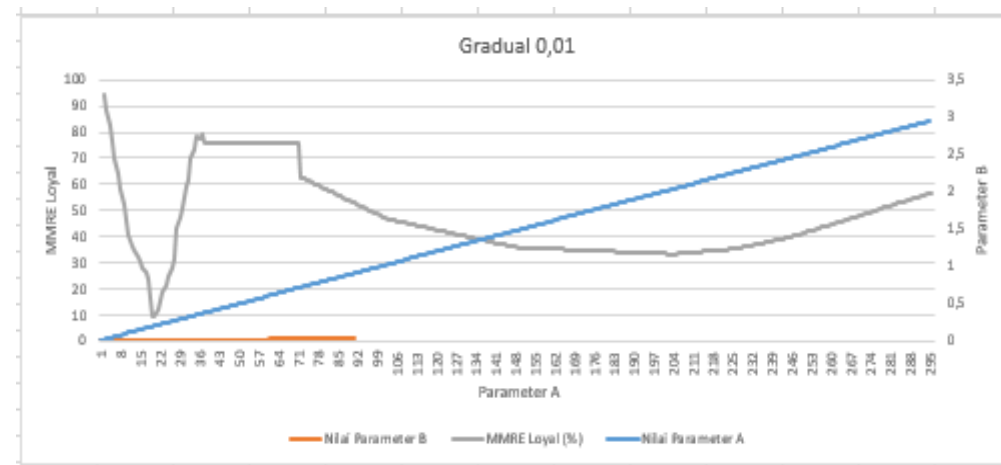
1. If (input1 is verylow) then (output1 is increasedsignificantly) (1)
2. If (input1 is low) then (output1 is increased) (1)
3. If (input1 is nominal) then (output1 is unchanged) (1)
4. If (input1 is high) then (output1 is decreased) (1)
5. If (input1 is veryhigh) then (output1 is decreasedsignificantly) (1)

LAMPIRAN 4 MMRE Loyal Dataset Cocomo-sdr Dengan Metode BCO

1. Tabel gradual 0,01 dengan 295 iterasi

Gradual 0,01				
Iterasi A	Nilai Parameter A	Iterasi B	Nilai Parameter B	MMRE Loyal (%)
A1	0,01	B1	0,01	94,1
A2	0,02	B2	0,02	88,2
A3	0,03	B3	0,03	82,2
A4	0,04	B4	0,04	76,3
A5	0,05	B5	0,05	70,3
A6	0,06	B6	0,06	64,4
A7	0,07	B7	0,07	58,5
A8	0,08	B8	0,08	52,6
A9	0,09	B9	0,09	46,6
A10	0,1	B10	0,1	40,7
A11	0,11	B11	0,11	36,6
A12	0,12	B12	0,12	34,5
A13	0,13	B13	0,13	32,4
A14	0,14	B14	0,14	30,3
A15	0,15	B15	0,15	28,2
A16	0,16	B16	0,16	26,1
A17	0,17	B17	0,17	23,9
A18	0,18	B18	0,18	9,71
A19	0,19	B19	0,19	9,21
A20	0,2	B20	0,2	12,3

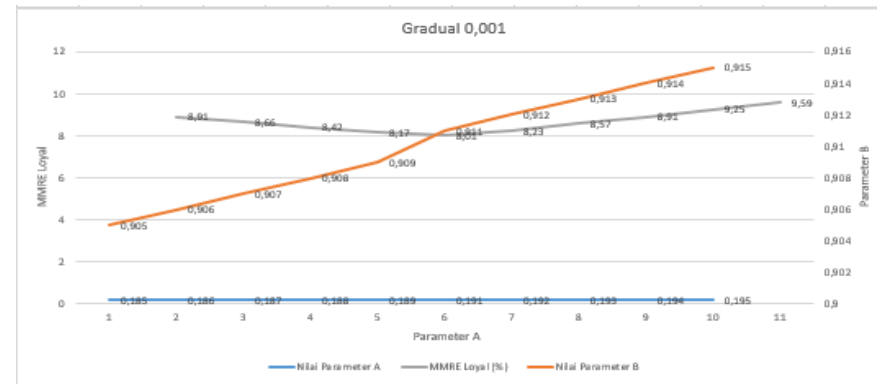
2. Grafik gradual 0,01 dengan 295 iterasi



3. Tabel gradual 0,001 dengan 10 iterasi

GRADUAL 0,001				
Iterasi A	Nilai Parameter A	Iterasi B	Nilai Parameter B	MMRE Loyal (%)
A1	0,185	B1	0,905	8,91
A2	0,186	B2	0,906	8,66
A3	0,187	B3	0,907	8,42
A4	0,188	B4	0,908	8,17
A5	0,189	B5	0,909	8,01
A6	0,191	B6	0,911	8,23
A7	0,192	B7	0,912	8,57
A8	0,193	B8	0,913	8,91
A9	0,194	B9	0,914	9,25
A10	0,195	B10	0,915	9,59

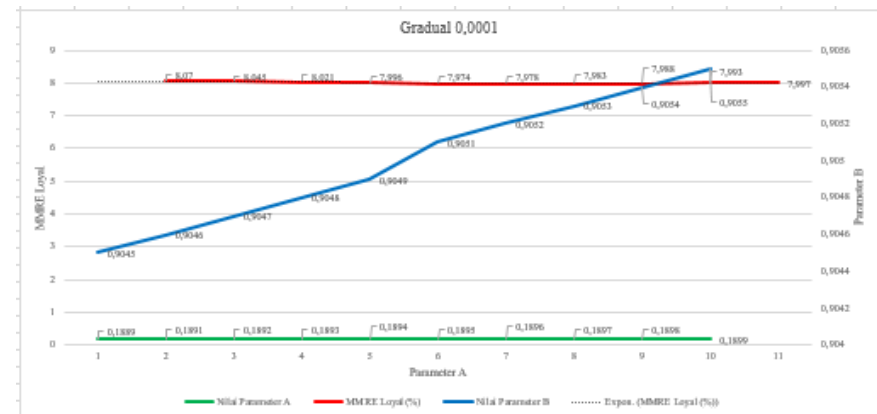
4. Grafik gradual 0,001 dengan 10 iterasi



5. Tabel gradual 0,0001 dengan 10 iterasi

GRADUAL 0,0001				
Iterasi A	Nilai Parameter A	Iterasi B	Nilai Parameter B	MMRE Loyal (%)
A1	0,1889	B1	0,9045	8,07
A2	0,1891	B2	0,9046	8,045
A3	0,1892	B3	0,9047	8,021
A4	0,1893	B4	0,9048	7,996
A5	0,1894	B5	0,9049	7,974
A6	0,1895	B6	0,9051	7,978
A7	0,1896	B7	0,9052	7,983
A8	0,1897	B8	0,9053	7,988
A9	0,1898	B9	0,9054	7,993
A10	0,1899	B10	0,9055	7,997

6. Grafik gradual 0,0001 dengan 10 iterasi

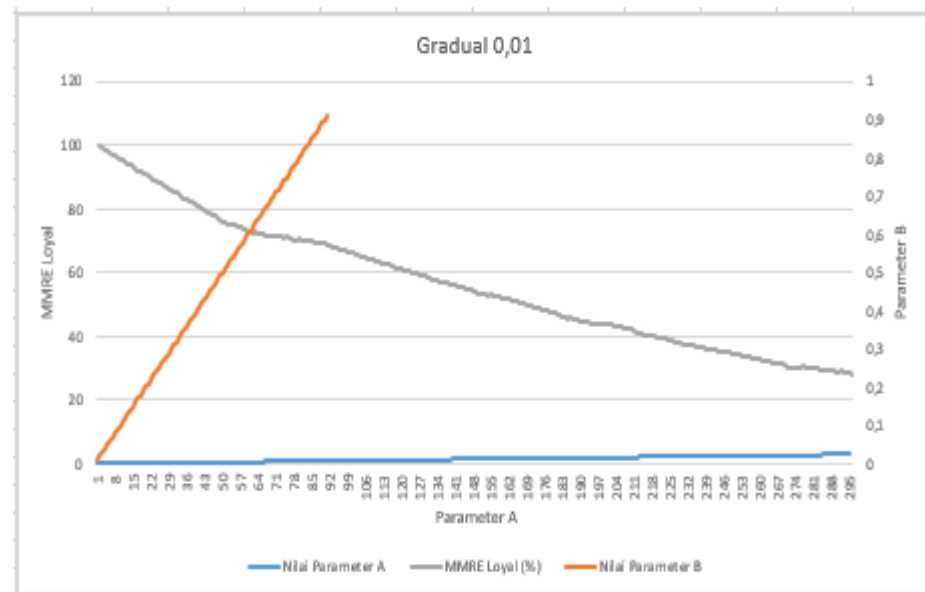


LAMPIRAN 5 MMRE Loyal Dataset Nasa 93-dem Dengan Metode BCO

1. Tabel gradual 0,01 dengan 295 iterasi

Gradual 0,01				
Iterasi A	Nilai Parameter A	Iterasi B	Nilai Parameter B	MMRE Loyal (%)
A1	0,01	B1	0,01	99,5
A2	0,02	B2	0,02	99
A3	0,03	B3	0,03	98,5
A4	0,04	B4	0,04	98,1
A5	0,05	B5	0,05	97,5
A6	0,06	B6	0,06	97,1
A7	0,07	B7	0,07	96,6
A8	0,08	B8	0,08	96,1
A9	0,09	B9	0,09	95,6
A10	0,1	B10	0,1	95,2
A11	0,11	B11	0,11	94,6
A12	0,12	B12	0,12	94,2
A13	0,13	B13	0,13	93,7
A14	0,14	B14	0,14	93,2
A15	0,15	B15	0,15	92,7
A16	0,16	B16	0,16	92,2
A17	0,17	B17	0,17	91,7
A18	0,18	B18	0,18	91,2
A19	0,19	B19	0,19	90,8
A20	0,2	B20	0,2	90,3

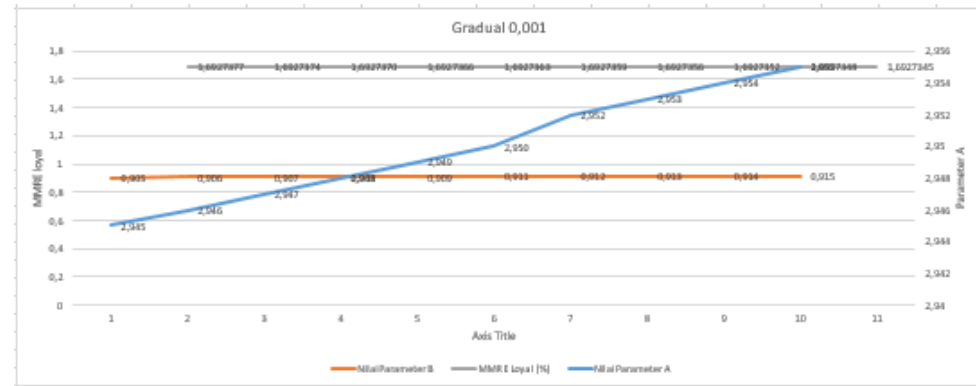
2. Grafik gradual 0,01 dengan 295 iterasi



3. Tabel gradual 0,001 dengan 10 iterasi

GRADUAL 0,001				
Iterasi A	Nilai Parameter A	Iterasi B	Nilai Parameter E	MMRE Loyal (%)
A1	2,945	B1	0,905	1,6927377
A2	2,946	B2	0,906	1,6927374
A3	2,947	B3	0,907	1,6927370
A4	2,948	B4	0,908	1,6927366
A5	2,949	B5	0,909	1,6927363
A6	2,950	B6	0,911	1,6927359
A7	2,952	B7	0,912	1,6927356
A8	2,953	B8	0,913	1,6927352
A9	2,954	B9	0,914	1,6927348
A10	2,955	B10	0,915	1,6927345

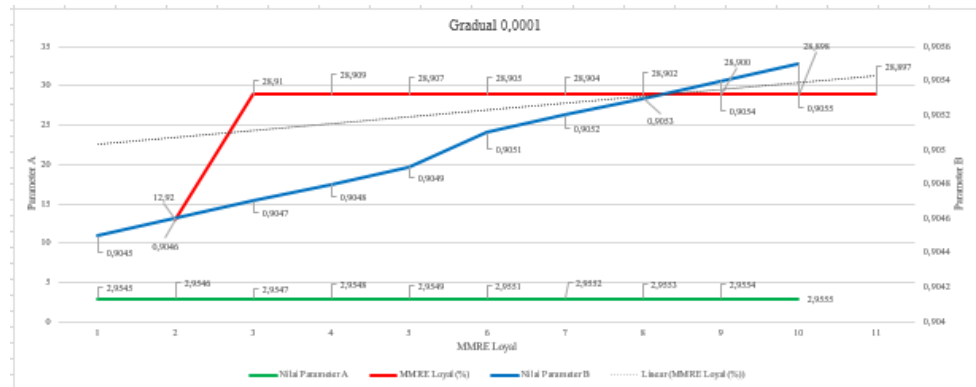
4. Grafik gradual 0,001 dengan 10 iterasi



5. Tabel gradual 0,0001 dengan 10 iterasi

GRADUAL 0,0001				
Iterasi A	Nilai Parameter A	Iterasi B	Nilai Parameter E	MMRE Loyal (%)
A1	2,9545	B1	0,9045	12,92
A2	2,9546	B2	0,9046	28,91
A3	2,9547	B3	0,9047	28,909
A4	2,9548	B4	0,9048	28,907
A5	2,9549	B5	0,9049	28,905
A6	2,9551	B6	0,9051	28,904
A7	2,9552	B7	0,9052	28,902
A8	2,9553	B8	0,9053	28,900
A9	2,9554	B9	0,9054	28,898
A10	2,9555	B10	0,9055	28,897

6. Grafik gradual 0,0001 dengan 10 iterasi



BIODATA PENULIS



Penulis, Rahmi Rizkiana Putri, lahir di kota Banjarmasin pada tanggal 18 November 1990. Penulis adalah anak ketiga dari tiga bersaudara dan dibesarkan di kota Banjarmasin, Kalimantan Selatan.

Penulis menempuh pendidikan formal di SD Negeri Pasar Lama 1 Banjarmasin (1996-2002) SMPN 2 Banjarmasin (2002-2005), dan SMK Negeri 2 Banjarmasin (2005-2008).

Pada tahun 2008-2011, penulis melanjutkan pendidikan D3 di Jurusan Teknik Informatika, Politeknik Negeri Banjarmasin. Pada tahun 2012-2014, penulis melanjutkan pendidikan D4 di Jurusan Teknik Informatika, Politeknik Elektronika Negeri Surabaya.

Pada tahun 2015-2017, penulis melanjutkan pendidikan Magister S2 di Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur. Di Jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak. Penulis dapat dihubungi melalui alamat email rahmirizkianaputri@gmail.com.